IDEAS FOR MSX

Edited by Ken Zetie

IDEAS

FOR

MSX

Introduction

This book will not teach you to program your MSX. It will, however, help both the 'learner' and the experienced programmer alike. It will also, I hope, provide you with hours of fun with the many games and graphics programs included.

The programs are all presented as listings direct from the MSX, reducing the chance of typographical errors.

Along with each program is a summary and a commentary. The summary describes the program - what it tries to achieve, the aim of the game, points arising from the program etc. In the commentary the programmer has broken the program down into sections and described what each part does, and to some extent, how it does it. This section is intended to allow more experienced programmers to try to follow how the program works. Using the outline presented, most people should be able to either use routines wholesale, or adapt them slightly for use in their own programs. With that in mind, many ideas which are repeated in a number of programs have been written in different ways - for example, the presentation of instructions in some programs is printed directly, in some the edges are justified, and in some, each line is centralised. Whichever method you prefer, there is an example somewhere in this book.

Even if you cannot understand all of the programs and the routines they use, the book may still give you plenty of ideas for your own programs, both in content and concept.

So, whether you find the contents of this book instructive, or just enjoyable, there should be plenty to excercise your mind for some time.

Ken Zetie
Editor

April 1985

Contents

Pom-Planets

Summary

Random shading is used in this program to generate the patterns. What happens is that the computer draws a circle, and as it does so the probability of a point being shaded changes. There is a small chance of the left hand side being coloured, but the center line will almost certainly be. This leads to a disrupted appearance, making them look somewhat like pom-pom balls, or planets in space (hence the title).

It is worth allowing a fair number of spheres to be drawn as it illustrates a problem with the hi-res screen. Where the spheres overlap the colour of one will 'leak' into the colour of the other. This is because the foreground and background colours are defined for groups of eight pixels, not individually, in order to save memory space. This means that when one pixel overlaps into another group which already contains two colours (fore and back), one of the colours in the second group is set to that of the intruding pixel.

For example, in the diagram below, pixels 11 and 13 are initially set to red. Then, when pixels 5 - 9 are set to yellow, 11 and 13 turn to yellow. If pixel 2 is then set to red, 5-7 are turned red. It can be seen therefore that the latest pixel set controls the colour of eight pixel blocks, always set as 0-7,8-15 etc.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Program Commentary

70-110   Sets up display screen (mode 2).

120-170   Seeds the random number generator with varying seed, then sets up the variables for the planet :

        E%, F% = x,y -co-ordinates.
        C% = color.
        S% = size.

180-240   Creates the planet as described above.

250-330   Prints a text message onto hi-res screen and repeats or stops the program as required.

```
10 REM *********************
20 REM **                **
30 REM ** POM POM PLANETS **
50 REM **                **
60 REM *********************
70 COLOR15,1,1 :SCREEN2,,1
80 OPEN"GRP:"AS#1
90 PRESET(80,130)
100 PRINT#1,"POM POM PLANETS"
110 LINE(80,140)-(196,140),15,B
120 T=TIME
130 E%=RND(T)*145+35
140 F%=RND(T)*60+35
150 C%=RND(T)*13+2
160 S%=RND(T)*35+1
170 D%=S%*S%
180 FORY%=-S%TOS%
190 X=SQR(D%-Y%*Y%)
200 FORI%=-XTO0
210 IFI%=0THENLINE(E%,F%+Y%)-(E%+X,F%+Y%),C%:GOTO240
220 IFRND(1)*X-X<I%THENCOLORC%,1,1ELSECOLOR1,1,1
230 PSET(E%+I%,F%+Y%)
240 NEXT:NEXT
250 COLOR15,1,1
260 PRESET(0,160)
270 PRINT#1,"    PRESS SPACE BAR TO CONTINUE"
280 PRESET(40,175)
290 PRINT#1,"     ANY OTHER TO END"
300 I$=INKEY$
310 IFI$=" "THENGOTO 120
320 IFI$<>""THENEND
330 GOTO300
```

Summary

Time for the clever stuff! Whereas the other programs produced static displays of 2D patterns, this little number produces not only a 3D picture, but rotates it as well!
The user is asked for the number of points on the base of the pyramid, and the computer will then display the picture after calculating tables of data necessary (this takes about 1 minute). You can then rotate the pyramid with pitch (up/down cursor keys), or roll (left/right keys).

Program Commentary

80-100	Set up screen and variables.
110-120	Input no. of points, check number is within range.
130-170	Calculate data and print wait message.
180-190	Define functions for X,Y co-ordinates of points on pyramid base.
200-220	Select hi-res screen and calculate Y coordinates of vertex and centre of base.
250-290	Fetch input, check for validity, and redirect control.
300	Repeat above section (from line 210)
330-410	Draws pyramid.
420-470	Compile Sin and Cos tables.
480-500	Centre text routine.
510-540	Print out of range error message (from line 110).

```
 10 REM ******************************
 20 REM * Rotating Pyramid Program *
 30 REM *          Written          *
 40 REM *    26th February 1985     *
 50 REM ******************************
 60 REM
 70 REM
 80 SCREEN0:KEYOFF
 90 S%=3:L%=80:H%=110:XS%=50:TH%=30
100 F!=45/ATN(1)
110 INPUT"Please enter the number of points on the base of the pyramid(2 to 10) ";S%
120 IFS%<2ORS%>10THENGOTO520
130 CLS
140 P$="Please wait for about a minute while":YP%=4:GOSUB490
150 P$="the computer calculates data used":YP%=9:GOSUB490
160 P$="later on in the program.":YP%=14:GOSUB490
170 GOSUB430
180 DEFFNX%(R%,U%)=128+U%*S!(R%MOD360)
190 DEFFNY%(R%,U%)=BY%+U%*C!(R%MOD360)
200 SCREEN2
210 YS%=XS%*SIN(TH%/F!)
220 TY%=96-50*C!(TH%):BY%=96+50*C!(TH%)
230 GOSUB320
240 BEEP
250 I$=INKEY$:IFI$<CHR$(28)ORI$>CHR$(31)THEN250
260 IFI$=CHR$(29)THENAN%=(AN%+6)MOD360:GOSUB320:GOTO240
270 IFI$=CHR$(28)THENAN%=AN%-6:IFAN%<0THENAN%=360+AN%
280 IFI$=CHR$(31)THENTH%=(TH%+6)MOD360:GOTO210
290 IFI$=CHR$(30)THENTH%=TH%-6:IFTH%<0THENTH%=360+TH%
300 GOTO210
310 END
320 CLS
330 XP%=FNX%(AN%,XS%):YP%=FNY%(AN%,YS%)
340 DRAW"BM128,=TY%;"
350 FORB%=AN%TO360+AN%STEP360/S%
360 XP%=FNX%(B%,XS%):YP%=FNY%(B%,YS%)
370 DRAW"M=XP%;,=YP%;"
380 DRAW"M128,=TY%;"
390 DRAW"BM=XP%;,=YP%;"
400 NEXT
410 RETURN
420 REM COMPILATION ROUTINE
430 DIMS!(360),C!(360)
440 FORA%=0TO360
450 S!(A%)=SIN(A%/F!):C!(A%)=COS(A%/F!)
460 NEXT
470 RETURN
480 REM CENTRED TEXT ROUTINE
490 LOCATE19-LEN(P$)/2,YP%:PRINTP$;
500 RETURN
510 REM OUTSIDE RANGE PROMPT
520 PRINT
530 PRINT"This number is outside the range of  2 - 10,try again"
540 PRINT:GOTO110
```

Trig Patterns

Summary

Let the computer do the work - this program shows how a complex pattern can be created by the computer using its inbuilt trig functions. The pattern is based on two circles whose radii are varied as they are swept out.

When the program is run the computer will take about one minute to calculate arrays of Sine and Cosine functions, in order to increase the running speed. Time to make a cup of coffee!

Program Commentary

80-130   Set up screen and display wait message (centered). Set up variables.

140-150   Set up hi-res screen.

160   Sets T% to 0 or 90.

190   Select sweep rates for the two circles.

200-270   Main loop to draw lines between two different circles as they are generated.

280-310   Rerun option.

330-390   Subroutine to store arrays containing Sine and Cos functions.

400-420   Text-centering routine (Screen 2).

430-450   Text-centering routine (Screen 0).

```
10 REM ********************************
20 REM ** PATTERN Generator Program **
30 REM **          Written          **
40 REM **        March  1985         **
50 REM ********************************
60 REM
70 REM
80 KEYOFF:COLOR15,4,4
90 SCREEN0
100 P$="Please wait for just over a  minute":Y%=6:GOSUB440
110 P$="while the computer calculates":Y%=12:GOSUB440
120 P$="data used later on in the program.":Y%=18:GOSUB440
130 GOSUB340
140 SCREEN2
150 OPEN"GRP:"AS#1
160 T%=-90*(RND(1)*11<6)
170 CLS
180 P$="MSX PATTERN GENERATOR":Y%=0:GOSUB410
190 F1%=RND(1)*16:F2%=RND(1)*16
200 FORA%=0TO360
210 R1%=90*S((A%*F1%)MOD360)
220 X%=128+R1%*S((A%+T%)MOD360):Y%=96+R1%*C((A%+T%)MOD360)
230 DRAW"BM=X%;,=Y%;"
240 R2%=90*C((A%*F2%)MOD360)
250 X%=128+R2%*S(A%):Y%=96+R2%*C(A%)
260 DRAW"M=X%;,=Y%;"
270 NEXT
280 BEEP
290 FORA%=0TO40:I$=INKEY$:NEXT:REM CLEAR KEYBOARD BUFFER
300 P$="Press SPACE BAR to run again.":Y%=182:GOSUB410
310 I$=INKEY$:IFI$<>" "THEN310
320 GOTO160
330 REM SIN & COS COMPILATION ROUTINE
340 DIMS(360),C(360)
350 F!=45/ATN(1)
360 FORA%=0TO360
370 S(A%)=SIN(A%/F!):C(A%)=COS(A%/F!)
380 NEXT
390 RETURN
400 REM CENTRED HI RES TEXT ROUTINE
410 PRESET(128-LEN(P$)*4,Y%):PRINT#1,P$;
420 RETURN
430 REM TEXT CENTERING ROUTINE
440 LOCATE20-LEN(P$)/2,Y%:PRINTP$;
450 RETURN
```

Curved Stitching Pattern

Summary

It is possible to produce curved patterns from straight lines - often this is done with needle and thread (hence the title) and sometimes by just drawing lines. This program, written at the request of the author's younger brother who had just discovered these in maths, emulates this procedure. It is a striking example of how to produce excellent patterns from a simple basis. The effect is enhanced by changing screen modes and colours.

Program Commentary

40-50   Set screen mode.

70-140   Loop to change colour for each picture.

80-130   Loop to draw picture.

150   Alter variable to control screen mode.

```
10 REM
20 REM CURVED STITCHING PATTERN
30 REM
40 S=1
50 SCREEN 2+S
60 COLOR 15,4,7
70 FORC=1TO15
80 FORR=0TO90STEP10
90 PSET(110,R+10),C:LINE-STEP(R,90-R),C
100 LINE-STEP(-R,90-R),C
110 LINE-STEP(-R,R-90),C
120 LINE-STEP(R,R-90),C
130 NEXT
140 NEXT
150 S=(S+1)MOD 2
160 GOTO 50
```

Polygon Patterns

Summary

Based on a simple polygon drawing routine this program produces patterns using rotating, expanding polygons of 2 to 10 sides. When each pattern is complete the basic polygon's name is printed at the top of the screen and the program waits for the space bar to be pressed before restarting with the next shape. To exit the program, type 'E' after the decagon.

Program Commentary

70	Calculates a degree-radian conversion factor. Divide by this to convert from degrees to radians.
90-100	Sets up screen 2 for output to allow text to be printed.
110-300	Comprise the main body of the program, calling subroutines elsewhere in the program. This structured type of programming is recomended, as programs written in this way are easier to follow and debug.
350-460	These are the subroutines called by the main program and other subroutines.
470-490	This routine will print the string P$ on the screen 2 display at vertical position YP%. It centres the text automatically.

```
10 REM *****************************
20 REM **                       **
30 REM ** Polygon Rotation Program **
40 REM **                       **
50 REM *****************************
60 REM
70 SCREEN0:COLOR15,0:KEYOFF:FACTR!=45/ATN(1)
80 GOSUB350:REM Call Initialisation Routine
90 SCREEN2,,0
100 OPEN"GRP:"AS#1:REM Open Hi Res Screen for Text Printing
110 FORSIDE%=2TO10
120 CLS
130 FORRADIUS%=5TO70STEP5
140 X%=(128+RADIUS%*S!(RADIUS%)):Y%=85+RADIUS%*C!(RADIUS%):DRAW"BM=X%;,=Y%;"
150 FORANGLE%=RADIUS%TORADIUS%+360STEP360/SIDE%
160 COLOR1+(ANGLE%-RADIUS%)/(360/SIDE%)+1,0,0
170 X%=128+RADIUS%*S!(ANGLE%)
180 Y%=85+RADIUS%*C!(ANGLE%)
190 DRAW"M=X%;,=Y%;"
200 NEXT
210 NEXT
-220 P$=D$(SIDE%-1):YP%=5
230 GOSUB470:REM Use Text Routine To Print Shape Type
240 FORE=0TO20:E$=INKEY$:NEXT:REM Ensure Keyboard Buffer Is Empty
250 IFSIDE%<10THENYP%=170:P$="Press SPACE to continue":GOSUB470
260 IFSIDE%<10THENI$=INKEY$:IFI$<>" "THEN260
270 NEXT
280 YP%=170:P$="SPACE To Repeat,E To End":GOSUB470
290 I$=INKEY$:IFI$<>" "ANDI$<>"E"ANDI$<>"e"THEN290
300 IFI$<>"E"GOTO110
310 KEYON:SCREEN0,,1
320 P$="Exit - ´GOTO 150´ Restarts Program":YP%=4:GOSUB520
330 COLOR15,0,0
340 END
350 DIMS!(430),C!(430),D$(9)
360 P$="Please Wait - Compiling SIN & COS":YP%=4
-370 GOSUB520:REM Use SCREEN0 Text Routine
380 P$="===========================":YP%=5:GOSUB520
390 YP%=15
400 FORA%=0TO430
410 P$=STR$(430-A%)+" ":GOSUB520
420 S!(A%)=SIN(A%/FACTR!):C!(A%)=COS(A%/FACTR!)
430 NEXT
440 RESTORE
450 FORA%=1TO9:READD$:D$(A%)=D$:NEXT
460 RETURN
470 COLOR15,0,0: PRESET(128-LEN(P$)*4,YP%):REM YP%=YCOORD
480 PRINT#1,P$
490 RETURN:REM P$=STRING TO BE PRINTED
500 DATA"LINE","TRIANGLE","SQUARE","PENTAGON","HEXAGON"
510 DATA"HEPTAGON","OCTAGON","NONAGON","DECAGON"
520 LOCATE20-LEN(P$)/2,YP%
530 PRINTP$;
540 RETURN
```

Line Patterns

Summary

Pretty pattern time again folks! This program generates patterns by moving the two ends of a line, bouncing them off the edges of the screen. Complex patterns can be formed in this way by what is essentially a simple program.

When running the program hitting the space bar will restart with new (random) variables.

Program Commentary

80-90   Select hi-res screen, open screen file for text output.

100   Reseeds random number generator using TIME variable, which gives a truly random number.

110-120 Select random step sizes for X and Y co-ordinates.

130-140 Select random start positions.

150-160 Use hi-res text centering routine (below) to print title and instructions.

170   Clear keyboard buffer.

180   Draw the line.

190-200 Add step sizes to X,Y co-ordinates.

210-240 Bounce line off edge of screen.

250   Repeat until user hits space bar.

260   Reruns program, new variables.

280-300 Hi-res text centering routine.

```
10 REM *************************
20 REM ** MSX LINE PATTERNS **
30 REM **       Written       **
40 REM **      March 1985      **
50 REM *************************
60 REM
70 REM
80 SCREEN2
90 OPEN"GRP:"AS#1
100 SEED%=RND(-TIME):REM RE-SEED RANDOM NUMBER GENERATOR
110 DA%=RND(1)*20:DB%=RND(1)*20:REM SET X & Y STEP SIZES
120 DC%=RND(1)*20:DD%=RND(1)*20
130 X1%=RND(1)*150+20:Y1%=RND(1)*150+20:REM SELECT RANDOM START POSITION
140 X2%=RND(1)*150+20:Y2%=RND(1)*150+20
150 P$="MSX LINE PATTERNS":Y%=1:GOSUB280
160 P$="Press SPACE BAR to run again":Y%=182:GOSUB280
170 FORA%=0TO40:I$=INKEY$:NEXT
180 LINE(X1%,Y1%)-(X2%,Y2%)
190 X1%=X1%+DA%:Y1%=Y1%+DB%:REM ADD STEP TO COORDINATES
200 X2%=X2%+DC%:Y2%=Y2%+DD%
210 IFX1%<20ORX1%>236THENDA%=-DA%:REM BOUNCE OFF SCREEN EDGE
220 IFY2%<20ORY2%>171THENDD%=-DD%
230 IFX2%<20ORX2%>236THENDC%=-DC%
240 IFY1%<20ORY1%>171THENDB%=-DB%
250 IFINKEY$<>" "THEN180
260 RUN
270 REM HI RES CENTRED TEXT ROUTINE
280 PRESET(128-LEN(P$)*4,Y%)
290 PRINT#1,P$
300 RETURN
```


MSX LINE PATTERNS
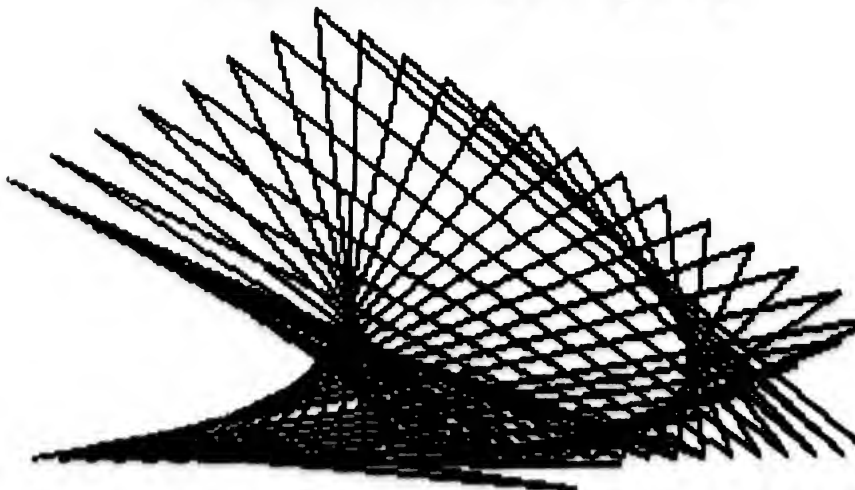
Press SPACE BAR to run

Star

Summary

First written on an RML380Z, then a BBC this program has a long history! The formula

Number = NP(NP-1)/2

gives the number of lines needed to join NP points. This program arranges the points at the corners of an NP sided polygon and uses a simple algorithm to 'join the dots'. When you've had enough use CTRL-STOP to exit the program.

Program Commentary

90-120   Set up arrays and screen.

130   Set F! as degree-radian convertor.

140-270   Main loop :

150   Calcuate angle between points in radians.

160-190   Sets positions of points.

200-210   Display number of points.

220-260   Two loops to draw the lines.

270   End main loop.

```
10 REM *****************************
20 REM ** 'STAR' Pattern Generator **
30 REM **         Program          **
40 REM **         Written          **
50 REM **        March  1985        **
60 REM *****************************
70 REM
80 REM
90 DIMX%(360),Y%(360)
100 SCREEN2
110 COLOR15,4,4:KEYOFF
120 OPEN"GRP:"AS#1
130 F!=45/ATN(1)
140 FORN%=2TO360
150 S!=360/N%/F!
160 FORA%=1TON%
170 X%(A%)=128+90*SIN(A%*S!)
180 Y%(A%)=96+90*COS(A%*S!)
190 NEXT
200 CLS
210 PRESET(0,0):PRINT#1,STR$(N%)+" POINTS"
220 FORA%=1TON%-1
230 FORB%=A%TON%
240 DRAW"BM=X%(A%);,=Y%(A%);M=X%(B%);,=Y%(B%);"
250 NEXT
260 NEXT
270 NEXT
280 END
```

9 POINTS
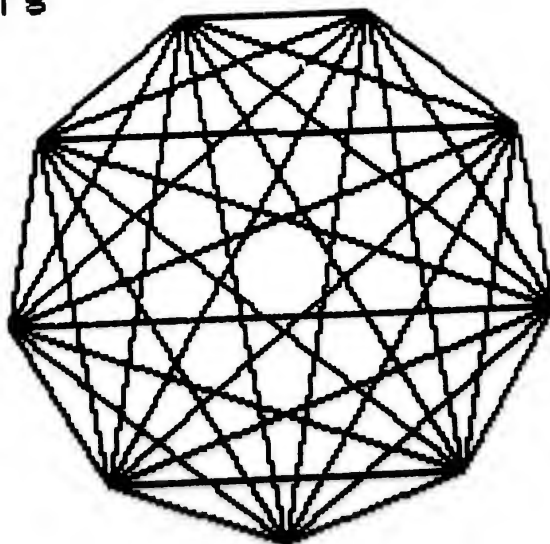
Missile

Summary

This is a game for those of you who remember the war. You are the commander of a submarine, firing torpedoes at ships of various sizes as they float gently across the horizon. However, tactics are involved as the different sizes are worth differing amounts of points, and are not equally difficult to destroy. Also, your supplies of missiles (torpedoes) is limited - shipping goods to war zones is not easy remember - so use them sparingly. The highest score to date was achieved by the author's younger brother, but this should not put people off as it is perhaps a more sedate game for the older generation (18+ say).

Program Commentary

40-60   Set up the screen for display.

70-100   Call the subroutine to generate the user-defined graphics, display the instructions, and create the ship strings.

110-890   This is the real meat of the program, as it controls the movement of your ship, and scrolls the ships across the screen.

870   Checks to see if the score has passed the 100's mark, and, if so, gives an extra missile, as promised in the instructions. (A really user-friendly program - it does not lie to you!)

```
10 REM
20 REM MISSILE
30 REM
40 OH=1
50 SCREEN 1
60 KEYOFF
70 GOSUB900
80 GOSUB630
90 GOSUB410
100 GOSUB440
110 HS=0
120 FORX=1TO100:J=0:GZ$=G$(INT(RND(1)*5+1))
130 FORXZ=1TO26
140 LOCATE0,5:PRINT"                              "
150 LOCATEXZ,5:PRINTGZ$
160 LOCATECOL,LIN:PRINT" "
170 K$="A":K$=INKEY$
180 IFCOL<=4 THEN COL=4
190 IFK$<>"Z"THEN GOTO 210
200 COL=COL-2
210 IFCOL>=36 THEN COL=36
220 IFK$<>"X"THEN GOTO 240
230 COL=COL+2
240 LOCATECOL,LIN:PRINT"+"
250 LOCATECOL,4:PRINT" "
260 IFK$<>"-" THEN GOTO 330
270 L=0:Y=20
280 EX=EX-1
290 LOCATECOL-1,5:PRINT"g"
300 LOCATECOL+1,5:PRINT"h"
310 BEEP:GOSUB 800
320 IFJ=1THEN GOTO 390
330 FORD=1 TO 10:NEXT
340 IFEX<=0 THEN GOTO 530
350 X$=CHR$(INT(RND(1)*6)+65)
360 PLAY"L32V9O2 XX$;"
370 NEXT XZ
380 LOCATE0,5:PRINT"                              "
390 NEXTX
400 GOTO 530
410 COLOR 14,4,7:LIN=20:COL=15:EX=10:PRINT
420 GOSUB880
430 RETURN
440 G$(1)="a       "
450 G$(2)="bc      "
460 G$(3)="def     "
470 G$(4)="def     "
480 FORX=5TO6
490 G$(X)="        "
500 NEXT X
510 SC=0
520 RETURN
530 SC$=STR$(SC):LOCATE6,10:PRINT"YOUR SCORE WAS"
540 LOCATE20,10:PRINTSC$:FORR=1TO50:NEXT
```

```
550  SC=0:EX=10
560  LOCATE3,16:PRINT"PRESS ANY KEY TO RESTART"
570  FORR=1TO10:A$=INKEY$:NEXT
580  A$=INKEY$:IFA$=""THENGOTO580
590  CLS
600  GOSUB410
610  GOSUB440
620  GOTO120
630  COLOR 1,4,7:CLS:PRINTSPC(8)" MISSILE ATTACK"
640  PRINTSPC(8)" --------------"
650  PRINT"THE OBJECT OF THE GAME IS TO"
660  PRINT"SINK THE ENEMY SHIPS WITH   "
670  PRINT"YOUR MISSILES AS THEY CROSS"
680  PRINT"THE HORIZON":PRINT"---------------------------"
690  PRINT:PRINTSPC(6)"Z = MOVE LEFT"
700  PRINT:PRINTSPC(6)"X = MOVE RIGHT"
710  PRINT:PRINTSPC(6)"- = FIRE MISSILE"
720  PRINT:PRINTSPC(6)" HIT ON def SCORES 10"
730  PRINT:PRINTSPC(6)" HIT ON bc  SCORES 20"
740  PRINT:PRINTSPC(6)" HIT ON a   SCORES 30"
750  PRINT" EVERY 100 POINTS EARNS A       MISSILE"
760  PRINT" PRESS ANY KEY TO START"
770  A$=INKEY$
780  A$=INKEY$:IFA$=""THENGOTO780
790  CLS:RETURN
800  GOTO810
810  IF VPEEK((32*6)+COL+6114)=97 THEN SC=SC+30:J=1
820  IF VPEEK((32*6)+COL+6114)=98 OR VPEEK((32*6)+COL+6114)=99 THEN SC=SC+20:J=1
830  VPE=VPEEK((32*6)+COL+6114):IFVPE=100ORVPE=101ORVPE=102THEN GOTO 850
840  GOTO 860
850  SC=SC+10:J=1
860  IF SC>HS THENHS=SC
870  IFSC/100>OH AND SC<>0 THEN EX=EX+1:OH=OH+1
880  LOCATE3,0:PRINT"SCORE:";SC;"MISSILES:";EX:PRINT"HIGH SCORE:";HS
890  RETURN
900  RESTORE:FOR U=(97*8) TO (105*8)-1
910  READG
920  VPOKEU,G
930  NEXT
940  DATA0,0,0,0,0,16,255,0
950  DATA0,0,0,0,1,39,255,127
960  DATA0,0,0,0,128,230,255,254
970  DATA0,0,0,0,1,255,127,63
980  DATA0,24,60,63,255,255,255,255
990  DATA0,0,0,0,192,255,254,252
1000 DATA8,4,2,63,63,2,4,8
1010 DATA8,16,32,63,63,32,16,8
1020 RETURN
```

Fox And Hounds

Summary

This is a version of the old game played on a chess board, and in fact was one of the first computerised games . In this version you play against the computer.

One player, in this case the MSX, has one piece which may be moved in any of the four diagonal directions, one square at a time - this is the fox. This player starts at the bottom row of the screen and it is his aim to reach the top row (for a can of Doggomeat or whatever). The other player, yourself, has four pieces (the hounds) which also move one square diagonally, but only move down the screen. Their aim is to trap the fox where it cannot move.

For those of you bored with chess and draughts this offers a worthwhile alternative, the new tactics involved requiring careful thought, as the odds of four to one are not as uneven as they sound.

Program Commentary

40-60   Set up the screen.

70-100   Direct the user to the instructions if required.

110-390   Set up the board for play.

400-410   Direct control to routine to find computer's move.

420-610   Generate the display. The colour codes for the characters are generated by means of the VPOKE commands in lines 430,440.

620-630   Checks if anyone has won at this stage.

640-770   Display the computer's move and input your reply. It then checks your move for validity.

780-1110   These control the fox's movement. This is done by checking randomly the possible positons to ensure that they are empty, and moving to a free one.

1230-1290 This section finishes the game with win/lose messages, and checks if another game is desired.

1300-1350 Display if the fox is trapped.

1360-1550 Print the instructions if required from lines 70-100.

```
10 REM
20 REM  FOX AND HOUNDS
30 REM
40 SCREEN 0
50 KEYOFF
60 COLOR 15,4,4
70 PRINT:PRINT:PRINT:PRINT"  INSTUCTIONS  (Y/N) ?"
80 FORR=1TO20:A$=INKEY$:NEXT
90 A$=INKEY$:IFA$=""THENGOTO90
100 IFA$="Y"THEN GOSUB 1360
110 DIML(9,9)
120 REM
130 OX=8:OY=5
140 CLS
150 FORC=1TO8STEP2
160 L(1,C)=1
170 L(1,C+1)=2
180 NEXT C
190 FORC=1TO8STEP2
200 L(2,C)=4
210 L(2,C+1)=1
220 L(3,C+1)=4
230 L(3,C)=1
240 L(4,C)=4
250 L(4,C+1)=1
260 L(5,C)=1
270 L(5,C+1)=4
280 L(6,C)=4
290 L(6,C+1)=1
300 L(7,C)=1
310 L(7,C+1)=4
320 L(8,C)=4
330 L(8,C+1)=1
340 NEXT C
350 L(8,5)=3
360 X=8
370 Y=5
380 A=X
390 B=Y
400 L(OX,OY)=4
410 GOSUB1300
420 SCREEN 0
430 VPOKE&H2000,1
440 VPOKE&H2001,33
450 CLS:PRINT:PRINT
460 LOCATE13,3:PRINT"FOX AND HOUNDS"
470 LOCATE13,4:PRINT"--------------"
480 PRINT"    12345678";
490 FORL=1TO8
500 PRINT:PRINTL;" ";
510 FORC=1TO8
520 IFL(L,C)=1THENPRINTCHR$(1)+"H";
530 IFL(L,C)=2THENPRINTCHR$(1)+"A";
540 IFL(L,C)=3THENPRINTCHR$(1)+"B";
```

```
550 IFL(L,C)=4THENPRINT" ";
560 NEXTC
570 PRINT" ";L;
580 NEXT
590 PRINT
600 PRINT"    12345678"
610 PRINT
620 IF X=1 THEN GOTO 1260
630 IF Z=1 THEN GOTO 1230
640 LOCATE0,22
650 PRINT"I MOVED FROM";OY;OX;"TO";Y;X
660 LOCATE0,17
670 PRINT" YOUR MOVE FROM (X,Y) ";
680 INPUTF1,F
690 PRINT"TO ";
700 INPUTT1,T
710 IFT<FTHENPRINT"YOU CANNOT MOVE BACKWARDS":GOTO660
720 Z=0
730 IFT-F<>1 THEN GOTO 760
740 IF L(F,F1)<>2 THEN GOTO 760
750 IFL(T,T1)=4 AND L(F,F1)=2 THEN GOTO 780
760 PRINT"INVALID MOVE"
770 GOTO 660
780 L(T,T1)=2
790 L(F,F1)=4
800 Q=0
810 K=INT(RND(1)*2)
820 OX=X:OY=Y
830 IFK>2 THEN GOTO 810
840 IFK=1 THEN GOTO 860
850 IFK=0 THEN GOTO 940
860 Q=Q+1
870 X=A-1
880 Y=B-1
890 IFL(X,Y)=4 THEN L(X,Y)=3
900 IFL(X,Y)<>3 AND Q<>2 THEN GOTO 940
910 IFL(X,Y)<>3 AND Q=2 THEN GOTO 1010
920 IFL(X,Y)=3 THEN L(X+1,Y+1)=4
930 IFL(X,Y)=3 THEN 400
940 X=X-1
950 Y=B+1
960 Q=Q+1
970 IFL(X,Y)=4 THEN L(X,Y)=3
980 IFL(X,Y)<>3 AND Q<2 THEN GOTO 860
990 IFL(X,Y)<>3 AND Q=2 THEN GOTO 1010
1000 IFL(X,Y)=3 THEN GOTO 400
1010 Z=0
1020 IFK=0 THEN GOTO 1040
1030 IFK=1 THEN GOTO 1120
1040 Q=Q+1
1050 X=A+1
1060 Y=B+1
1070 IFL(X,Y)=4 THEN L(X,Y)=3
1080 IFL(X,Y)<>3 AND Q<4 THEN GOTO 1120
```

```
1090 IFL(X,Y)=3 THEN L(X-1,Y-1)=4:GOTO 400
1100 IFL(X,Y)<>3 THEN Z=1
1110 GOTO 400
1120 REM
1130 Y=B-1
1140 X=A+1
1150 REM
1160 Q=Q+1
1170 IFL(X,Y)=4 THEN L(X,Y)=3
1180 IFL(X,Y)<>3 AND Q<4 THEN GOTO 1040
1190 IFL(X,Y)=3 THEN L(X-1,Y+1)=4
1200 IFL(X,Y)=3 THEN GOTO 400
1210 IFL(X,Y)<>3 THEN Z=1
1220 GOTO 400
1230 REM
1240 PRINT"  YOU WIN.....PLAY AGAIN   ";
1250 GOTO1270
1260 PRINT"   I  WIN.....PLAY AGAIN   ";
1270 INPUT"(YES OR NO)";A$
1280 IFA$="YES" THEN GOTO 120
1290 END
1300 REM
1310 IFA=X THEN GOTO 1350
1320 IFZ=1 THEN PRINT"I CAN´T MOVE":GOTO 1350
1330 A=X
1340 B=Y
1350 RETURN
1360 REM
1370 CLS
1380 PRINT:PRINT"   FOX AND HOUNDS"
1390 PRINT"   --------------"
1400 PRINT:PRINT
1410 PRINT" YOU ARE THE HOUNDS."
1420 PRINT" YOU HAVE FOUR PLAYERS (H)."
1430 PRINT" YOU CAN MOVE FORWARD ONLY."
1440 PRINT
1450 PRINT" I,THE COMPUTER,AM THE FOX."
1460 PRINT" I CAN MOVE BACKWARDS AND FORWARDS."
1470 PRINT
1480 PRINT" IF YOU TRAP ME YOU WIN."
1490 PRINT
1500 PRINT" IF I GET TO LINE 1 I WIN."
1510 PRINT
1520 PRINT"   TYPE RETURN TO CONTINUE."
1530 INPUTA$
1540 CLS
1550 RETURN
1560 A$=INKEY$:IFA$<>""THENPRINTASC(A$):GOTO 1560 ELSE 1560
```

Mutant Attack

Summary

    The year: 2053
    The place: A small island just off Brazil

    The situation is this: one of the post-holocaust survey teams is stranded after their hovercraft crash landed. You are the sole survivor, and find yourself on this small, isolated island. Dotted around the island are small, radioactive puddles. While observing one of these, something below stirs. You jump back just in time to avoid the thing which emerges in a flurry of radioactive liquid. It was not alone either. Soon from all around you zombie-like figures emerge from their pits and converge on you. Their only weakness seems to be reentering the pools...
    You take it in turns to move, but you are outnumbered by the monsters, and so careful forethought is required.
    Good luck, you'll need it!

Program Commentary

90-130   Call instructions if required.

140-210   Set up program to play.

220-290   Display the status - whose move, deaths etc.

300-310   General routine to display monsters and men.

360-470   Display the board, with creatures and scores.

480-730   Position monsters, moved in your general direction.

740-920   Input your move, checks for terminal illnesses, e.g death due to supplementing a monster's breakfast.

930-1090   Controls the monsters movement, checks for their demise.

1190-1310 Wind up the game after your death, or in the unlikely event of your winning.

```
10 REM
20 REM MUTANT ATTACK
30 REM
40 SCREEN 0
50 KEYOFF
60 COLOR 15,4,4
70 INPUT" ENTER RANDOMIZE FACTOR (1-100)";I
80 IFI<1ORI>100 THEN GOTO 70
90 PRINT:PRINT:PRINT:PRINT
100 PRINT" INSTRUCTIONS (Y/N) ?":FORF=1TO20:R$=INKEY$:NEXT
110 R$=INKEY$
120 IFR$="Y"THEN GOTO 1410
130 IFR$<>"N" THEN GOTO 110
140 SCREEN 0
150 GOSUB 1320
160 WIDTH 38
170 R=2
180 FORN=0 TO I
190 GOSUB 220
200 NEXT
210 GOTO 350
220 RN=1+INT(RND(1)*R)
230 IF RN<RM THEN GOTO 220
240 RETURN
250 LOCATE6,15:PRINT"                    "
260 LOCATE6,15:PRINTM$
270 PLAY"V9L3O3AB"
280 FOR PAUSE=1 TO 60:NEXT
290 RETURN
300 LOCATEPX,PY:PRINTM$
310 RETURN
320 I$=INKEY$
330 IFI$="" THEN GOTO 320
340 RETURN
350 CLS
360 PX=0:PY=0:M$="YOUR":GOSUB300
370 PX=26:PY=0:M$="MUTANT":GOSUB300
380 PX=0:PY=1:M$="SCORE":GOSUB300
390 PX=26:PY=1:M$="SCORE":GOSUB300
400 PX=0 :PY=2:M$=STR$(YS):GOSUB300
410 PX=26:PY=2:M$=STR$(MS):GOSUB300
420 PX=5:PY=0:M$=A$:GOSUB 300
430 PX=5:PY=14:M$=A$:GOSUB 300
440 FOR N=1TO13
450 PX=5:PY=N:M$=E$:GOSUB 300
460 PX=24:PY=N:M$=B$:GOSUB 300
470 NEXT N
480 R=30:RM=15
490 GOSUB 220
500 NP=RN
510 M$=CHR$(248)
520 FORN=1TONP
530 R=13:RM=0
540 GOSUB 220:PY=RN
```

```
550 R=23:RM=6
560 GOSUB220:PX=RN
570 GOSUB 300
580 NEXT N
590 R=15:RM=5
600 GOSUB 220:NM=RN
610 FORN=1 TO NM
620 R=13:RM=0
630 GOSUB 220:Y(N)=RN
640 R=23:RM=6
650 GOSUB 220:X(N)=RN
660 PX=X(N):PY=Y(N)
670 M$="M":GOSUB 300
680 NEXT N
690 LM=NM
700 R=13:RM=0
710 GOSUB 220:YY=RN
720 R=23:RM=6
730 GOSUB 220:YX=RN
740 S$=CHR$(VPEEK((40*YY)+YX+1))
750 IFS$<>" " THEN GOTO 700
760 PX=YX:PY=YY
770 M$="*":GOSUB 300
780 M$="YOUR MOVE"
790 GOSUB 250
800 GOSUB 320
810 PX=YX:PY=YY:M$=" ":GOSUB 300
820 IF I$="R" OR I$="D" OR I$="C" THEN YX=YX+1
830 IF I$="W" OR I$="S" OR I$="Z" THEN YX=YX-1
840 IF I$="W" OR I$="E" OR I$="R" THEN YY=YY-1
850 IF I$="Z" OR I$="X" OR I$="C" THEN YY=YY+1
860 S$=CHR$(VPEEK((YY*40)+YX+1))
870 IFS$=" " THEN GOTO 930
880 IFS$=CHR$(248) THEN M$="YOU FELL INTO A PUDDLE"
890 IFS$="M" THEN M$="RIGHT INTO A MUTANT"
900 IFS$=B$ THEN M$="YOU FELL INTO THE OCEAN"
910 GOSUB 250
920 GOTO 1150
930 PX=YX:PY=YY:M$="*":GOSUB 300
940 M$="HERE THEY COME!        "
950 GOSUB 250
960 FORN=1 TO NM
970 IF X(N)=99 THEN GOTO 1090
980 PX=X(N):PY=Y(N)
990 M$=" ":GOSUB 300
1000 IFX(N)<YX THEN X(N)=X(N)+1
1010 IFX(N)>YX THEN X(N)=X(N)-1
1020 IFY(N)<YY THEN Y(N)=Y(N)+1
1030 IFY(N)>YY THEN Y(N)=Y(N)-1
1040 S$=CHR$(VPEEK((40*Y(N))+X(N)+1))
1050 IFS$=CHR$(248)THEN M$="SPLASH.1 LESS MUTANT":X(N)=99:LM=LM-1:GOSUB250
1060 IFS$="*" THEN M$="CAUGHT YOU !!!":GOSUB 250:GOTO 1150
1070 PX=X(N):PY=Y(N):M$="M"
1080 IFS$=" "THEN GOSUB 300
```

```
1080 IFS$=" "THEN GOSUB 300
1090 NEXT N
1100 IF IM>0 THEN GOTO 780
1110 M$="  YOU WIN !!!"
1120 YS=YS+1
1130 GOSUB 250
1140 GOTO 1190
1150 M$="YOU´RE DEAD !!          "
1160 FOR PAUSE=1 TO 100:NEXT
1170 GOSUB 250
1180 MS=MS+1
1190 PRINT
1200 INPUT" ANOTHER GAME (YES OR NO)";I$
1210 IFI$="YES" THEN GOTO 350
1220 IFI$<>"NO" THEN GOTO 1200
1230 CLS
1240 PRINT:PRINT:PRINT" GOOD BYE !!!"
1250 PRINT
1260 PRINT"  YOUR SCORE ";YS
1270 PRINT
1280 PRINT"  MONSTERS SCORE ";MS
1290 PRINT
1300 PRINT
1310 END
1320 DIM X(50),Y(50)
1330 N=0:A$=""
1340 B$=CHR$(249)
1350 FORN=1 TO 20
1360 A$=A$+B$
1370 NEXT N
1380 MS=0:YS=0
1390 RM=0
1400 RETURN
1410 SCREEN1:PRINT
1420 PRINT"  MUTANT CHASE"
1430 PRINT
1440 PRINT" YOU ARE THE   *"
1450 PRINT
1460 PRINT" TO STEER USE..."
1470 PRINT
1480 PRINT"          E"
1490 PRINT"       W   R"
1500 PRINT"     S  ";CHR$(1)+"O    D"
1510 PRINT"       Z   C"
1520 PPINT"          X"
1530 PRINT
1540 PRINT" DON´T FALL IN THE PUDDIES "+CHR$(248)
1550 PRINT" OR INTO THE SEA "+CHR$(249)
1560 PRINT
1570 PRINT" PRESS ANY KEY TO START"
1580 FORR=1 TO 50:A$=INKEY$:NEXT
1590 A$=INKEY$:IF A$="" THEN GOTO 1590
1600 GOTO 140
1610 INPUTA$:PRINTCHR$(1)+A$:GOTO 1610
```

Pong

Summary

Not a character from Puccini´s Turandot, but an exercise in nostalgia. Remember the good old days before Space Invaders when pubs started filling with electronic Ping-Pong. Crude and unimaginative they are considered now, but back then they were the greatest thing since sliced chips (of the silicon variety). Now you can once more enjoy a game where no-one gets killed or maimed.

Although it is written for two players, it can be modified for one as shown below (listing 3). If you do not own a set of joysticks, you are advised to get some as this game, and others are much more playable with them. Listing 2 gives the amendments necessary for joystick control.

Program Commentary

80-130   Set up characters.

470-560   Draw screen outline.

650-680   Set variables for start of game.

690-770   Moves ball, checks for striking edge, prints scores.

780-790   Update time, check for end of game.

800-880   Move bats. Reprints sections of screen when cleared.

930-960   Flashes screen.

```
10 REM
20 REM PONG
30 REM
40 SCREEN 0
50 COLOR 3,1,1
60 KEYOFF
70 REM UDGs
80 RESTORE
90 FORR=97TO98:FORT=0TO7
100 READA$:VPOKE(R*8)+T+2048,VAL("&B"+A$+"00")
110 NEXTT,R
120 DATA000000,001100,011110,111111,111111,011110,001100,000000
130 DATA001100,001100,001100,001100,001100,001100,001100,001100
140 CLS
150 PRINT:PRINT"   PONG"
160 PRINT"   ----"
170 PRINT
180 PRINT"   THIS IS A VERSION OF THE TRADITIONAL"
190 PRINT" ARCADE GAME- PONG."
200 PRINT
210 PRINT" THIS IS A TWO PLAYER GAME "
220 PRINT
230 PRINT" PLAYER        UP    DOWN "
240 PRINT" ------        ----  ---- "
250 PRINT
260 PRINT" 1 (LEFT)      A     Z   "
270 PRINT
280 PRINT" 2 (RIGHT)     :     /   "
290 PRINT
300 PRINT
310 PRINT
320 PRINT"   PRESS ANY KEY TO CONTINUE"
330 FORR=1TO10:A$=INKEY$:NEXT
340 A$=INKEY$:IFA$=""THENGOTO340
350 CLS
360 PRINT:PRINT
370 PRINT" THE BALL WILL BOUNCE OF THE TOP"
380 PRINT"AND BOTTOM WALLS BUT IF IT HITS"
390 PRINT"EITHER OF THE END WALLS THE PLAYER"
400 PRINT"WHO BOUNCED IT THERE EARNS THE POINTS"
410 PRINT
420 PRINT"THE GAME CONTINUES UNTIL THE TIMER"
430 PRINT"REACHES ZERO."
440 PRINT
450 INPUT" HOW MANY SECONDS DO YOU WISH TO PLAY  FOR ";SE
460 IFSE<1THENGOTO350
470 CLS
480 PRINT
490 COLOR 1,3,7
500 PRINT" ------------------------------------"
510 FORR=1TO19
520 PRINT" |                                  |"
530 NEXT
540 PRINT" ------------------------------------"
```

```
550 LOCATE 4,0:PRINT"TIME ";SE
560 LOCATE 5,22:PRINT"0":LOCATE30,22:PRINT"0"
570 LOCATE10,10:PRINT" PRESS SPACE TO START"
580 A$=INKEY$
590 IF A$<>" "THENGOTO580
600 LOCATE10,10:PRINT"                    "
610 FORR=7 TO 1 STEP-1:X$=CHR$(64+R)
620 PLAY"L32O3V9 XX$;"
630 NEXT
640 FORR=1 TO 100:NEXT
650 A$=" ":FORR=1TO4:A$=A$+CHR$(29)+CHR$(31)+"b":NEXT:A$=A$+CHR$(29)+CHR$(31)+" "
660 XL=5:XR=33:YL=10:YR=10:A=TIME:TIME=0
670 BX=18:BY=INT(RND(1)*10)+5
680 Q=RND(-A)
690 DX=INT(RND(1)*2):IFDX=0THENDX=-1
700 DY=INT(RND(1)*2):IFDY=0THENDY=-1
710 LOCATEBX,BY:PRINT" "
720 IF BY+DY=1 OR BY+DY=21 THENDY=-DY
730 IF BX=2 THEN SR=SR+1:LOCATE30,22:PRINTSR:LOCATEBX,BY:PRINT"|":GOTO910
740 IFBX=36 THEN SL=SL+1:LOCATE5,22:PRINTSL:LOCATEBX,BY:PRINT"|":GOTO910
750 BX=BX+DX:BY=BY+DY
760 IF VPEEK((40*BY)+BX+1)=98THENDX=-DX
770 LOCATEBX,BY:PRINT"a"
780 TI=SE-INT(TIME/50):LOCATE9,0:PRINTTI;"  "
790 IFTI<=0 THEN GOTO 990
800 B$=INKEY$
810 IF(B$="A"ORB$="a")ANDYL<>1 THEN YL=YL-1
820 IF(B$="Z"ORB$="z")ANDYL<>16 THENYL=YL+1
830 IF B$=":"ANDYR<>1 THEN YR=YR-1
840 IF B$="/"ANDYR<>16 THENYR=YR+1
850 LOCATEXL,YL:PRINTA$
860 LOCATEXL,1:PRINT"-":LOCATEXL,21:PRINT"-"
870 LOCATEXR,YR:PRINTA$
880 LOCATEXR,1:PRINT"-":LOCATEXR,21:PRINT"-"
890 PLAY"L32V9O1 A"
900 GOTO 710
910 PLAY"V12O5 GAGA"
920 TT=TIME
930 FORR=1 TO 10
940 COLOR11,R,7
950 FORT=1TO15
960 NEXTT,R
970 TIME=TT
980 COLOR 1,3,7:GOTO670
990 CLS:PLAY"V15L40 GAGA R10 BABA"
1000 PRINT:PRINT
1010 PRINT"  TIME  UP"
1020 PRINT
1030 PRINT"  THE SCORES WERE..."
1040 PRINT
1050 PRINT" PLAYER 1 - LEFT     ";SL
1060 PRINT
1070 PRINT" PLAYER 2 - RIGHT    ";SR
1080 PRINT
```

```
1090 IFSR=SLTHENPRINT"    A  DRAW"
1100 IFSL>SRTHENPRINT" PLAYER 1 IS THE WINNER"
1110 IFSL<SRTHENPRINT" PLAYER 2 IS THE WINNER"
1120 PRINT
1130 PRINT
1140 FORR=1TO50:A$=INKEY$:NEXT
1150 PRINT"   PRESS SPACE TO START AGAIN"
1160 A$=INKEY$:IFA$<>" "THENGOTO1160
1170 SR=0:SL=0
1180 GOTO10
```

JOYSTICK AMENDMENTS

```
800 Al=STICK(1):A(2)=STICK(2)
810 IFAl=1ANDYL<>1 THEN YL=YL-1
820 IFAl=5ANDYL<>16 THENYL=YL+1
830 IFA2=1ANDYR<>1 THEN YR=YR-1
840 IFA2=5ANDYR<>16 THEN YR=YR+1
```

SINGLE PLAYER AMENDMENTS

```
210 REM
280 REM
52C PRINT"   |                            b"
660 XL=5:YL=10::A=TIME:TIME=0
730 IF BX=2 THEN LOCATEBX,BY:PRINT"|":SR=SR+1:LOCATE30,22:PRINTSR:GOTO910
740 IFBX=35 THEN DX=-DX
830 REM
840 REM
870 REM
880 REM
1030 REM
1050 PRINT" YOU MISSED ";SR;" BALL";:IFSR<>1THENPRINT"S"
1070 REM
1090 REM
1100 REM
1110 REM
```

Gunner

Summary

You are in charge of a 25-pounder dug in on one side of the river. On the other side, dug in and secure in their trench are the enemy (fill in name of enemy on dotted line...). Unfortunately your gun barrel will not descend below 30 degrees, so when it comes to aiming at them, you have to judge a tricky angle. You get as many goes as you need to hit the enemy trench, and your rating is based on number of shots needed – hit first time and become a "master gunner"!

When you hit, a new (random) position is chosen and you try again. Definitely a game for the thinker and tactician rather than the nimble fingered 'shoot the wombat' arcade game type!

To play the game, run it and you are presented with a screen, and the word 'ANGLE' at the top. To change the firing angle of the gun use 'A' (+5 degrees), 'Z' (-5), 'S' (+1), and 'X' (-1). When you have the angle you require, the gun is fired by pressing the return key, and you can abandon the game by pressing the space bar.

Program Commentary

10-90   Control general program flow, calling routines below.

120   Defines a function to convert degrees to radians.

210-280 Set graphics strings to draw ground, trench and barrel.
         Set variables and position of trench.

250   This sets the random number generator seed. The RNG
         actually consists of a list of numbers which it uses
         sequentially. Setting a dummy variable to RND(-x),
         where x is any number, resets the seed, i.e. where in
         the list it takes numbers from. Using the time variable
         as the seed creates a truly random number.

290-390 This subroutine sets the screen up, drawing the gun,
         ground trench and barrel.

500-570 Input user's command – move barrel or fire accordingly.

550   This draws a box in the background colour around the
         angle, thus clearing it, as plotting a space over text
         printed in this mode has no effect.

580-670 Plots the trajectory using the equation in line 590. If
         the trench is hit, the flag H is set to 1 and the
         explosion is drawn (line 680). The VDP commands make
         the screen display shudder.

740-900 End the game, display rating.

```
10 REM gunner
20 GOSUB 120
30 GOSUB 210
40 GOSUB 290
50 GOSUB 400
60 GOSUB 580
70 IF H=1 THEN GOTO 100
80 TR=TR+1
90 GOTO 40
100 GOSUB 740
110 CLOSE#1:GOTO30
120 DEFFNR(J)=ATN(1)/45*J
130 DIMS(85),C(85)
140 CLS
150 LOCATE8,10:PRINT"Please Wait"
160 FOR A=30 TO 85
170 S(A)=SIN(FNR(A))
180 C(A)=COS(FNR(A))
190 NEXT
200 RETURN
210 G$="bm0,180 r50d2r100u2r116"
220 T$="br5d3l10u3"
230 B$="bm25,170 M+=x;,-=y;"
240 L=20:V=45:H=0
250 TR=1:QQ=RND(-TIME)
260 TX=156+INT(RND(1)*95)
270 OPEN"grp:" AS #1
280 RETURN
290 SCREEN2:COLOR 15,4,7
300 AN=45
310 DRAW G$
320 LINE(50,182)-(150,192),,BF
330 PSET(TX,180)
340 DRAW T$
350 CIRCLE(25,170),10,,,,1.15
360 PRESET(10,10):PRINT#1,"Angle ";AN
370 X=L*C(AN):Y=L*S(AN)
380 DRAW B$
390 RETURN
400 AN$=INPUT$(1)
410 AO=AN
420 IF AN$="A" OR AN$="a" THEN AN=AN+5
430 IF AN$="Z" OR AN$="z" THEN AN=AN-5
440 IF AN$="S" OR AN$="s" THEN AN=AN+1
450 IF AN$="X" OR AN$="x" THEN AN=AN-1
460 IF AN$=" "THEN CLOSE#1:GOTO 30
470 IF AN<30 THEN AN=30
480 IF AN>85 THEN AN=85
490 IF ASC(AN$)=13 THEN RETURN
500 X=L*C(AO):Y=L*S(AO)
510 DRAW"c4 xb$;"
520 PSET(25+X/2,170-Y/2)
530 X=L*C(AN):Y=L*S(AN)
540 DRAW"c15 xb$;"
```

```
550 LINE(94,18)-(58,10),4,BF
560 PRINT #1,AN
570 GOTO 400
580 X=0
590 Y=-(X*S(AN)/C(AN)-((5/(V*V))*X*X/(C(AN)*C(AN))))
600 XP=25+X+L*C(AN):YP=170+Y/2-L*S(AN)
610 PSET(XP,YP)
620 IF YP>=177 OR XP>255 THEN GOTO 650
630 X=X+5
640 GOTO 590
650 IF ABS(TX-XP)<=5 THEN H=1
660 GOSUB 680
670 RETURN
680 DRAW"nul0nel0nrl0nfl0ndl0ngl0nll0nhl0"
690 SOUND7,7
700 PLAY"lln30"
710 VDP(0)=1:FORQQ=1TO100:NEXT:VDP(0)=2
720 FORQQ=1TO600:NEXT
730 RETURN
740 SCREEN 0:WIDTH 36
750 KEYOFF
760 LOCATE11,1:PRINT"Congratulations"
770 LOCATE1,3:PRINT"You hit the enemy trench after only";TR;" attempt";
780 IF TR<>1 THEN PRINT"s"
790 IF TR=1 THEN R$="Master Gunner":GOTO 860
800 IF TR=2 THEN R$="Very Good":GOTO 860
810 IF TR=3 THEN R$="Good":GOTO 860
820 IF TR<=5 THEN R$="Average":GOTO 860
830 IF TR<=10 THEN R$="Poor":GOTO 860
840 IF TR<=15 THEN R$="Atrocious":GOTO 860
850 R$="Give up now"
860 LOCATE1,6:PRINT"Rating ";R$
870 LOCATE1,10:PRINT"Press Space Bar"
880 ZZ$=INPUT$(1)
890 IF ZZ$<>" " THEN GOTO 870
900 RETURN
```

Post-Office Game

Summary

This game is perhaps better known as 'Mastermind' since its publication by Invicta. Here, the computer sets you a code of four numbers and you have to crack it in as few goes as possible. You get a maximum of 15 attempts. Inputting your guess is simple - the computer prints a dot in the position of the next guess and you merely type the number you want , from 1 to 5. Once you have entered four numbers, you enter the guess by pressing return. At any time, pressing 'D' will delete the previous entry.

The computer then marks your guess in the following manner: For each correct number in the correct position you get a '+'. For each correct number but in the wrong position, a '-'.

For example a mark of '+ - - ' would mean that your guess contained three of the numbers in the original code, and one of them in the right place. NB If there were one 5 in the answer, this mark would be achieved for three 5's in the guess.

Program Commentary

30-90   These lines are the control loop, calling subroutines for inputting and marking guesses.

100-190   The initialisation routine, this sets the screen, arrays for answer and guess, and also sets the code to guess.

200-280   This is the input routine, and builds up the array AT( ) containing the guessed numbers. It can only be exited by pressing return (line 270).

300-440   The marking routine is quite complex, and difficult to follow. It builds up E$ which is the final mark by first counting the number of plusses and then the number of minuses, not duplicating the plusses. It does the check in two passes so as not to give the player any clues as to which numbers the marks refer to.

450-540   This is the closing routine which prints an appropriate message, and asks if you want to play again.

```
10 REM Post-office game
20 GOSUB 100
30 FOR I%=0 TO 15
40 GOSUB 200
50 GOSUB 290
60 IF E=1 THEN I%=15
70 NEXT
80 GOSUB 450
90 RUN
100 SCREEN 0:KEYOFF:WIDTH 36
110 COLOR 15,4,7:DIM AN(4),AT(4):QQ=RND(-TIME)
120 FOR J%=1 TO 4
130 AN(J%)=INT(RND(1)*5)+1
140 NEXT
150 LOCATE10,1,0:PRINT"Post-Office Game"
160 FOR J%=2 TO 8 STEP 2
170 LOCATE J%,3:PRINT"*"
180 NEXT
190 RETURN
200 J%=1
210 LOCATE(J%*2)+17*(I%\8),5+2*(I%MOD8)
220 PRINT".";CHR$(&H1D);
230 IN$=INPUT$(1):IN=VAL(IN$)
240 IF J%=5 THEN GOTO 260
250 IF IN>0 AND IN<6 THENAT(J%)=IN:PRINTCHR$(&H1D);STR$(IN);:J%=J%+1:GOTO210
260 IF (IN$="D" OR IN$="d") AND J%<>1 THEN J%=J%-1:GOTO210
270 IF ASC(IN$)=13 AND J%=5 THEN RETURN
280 GOTO 230
290 E$="":E=0
300 P=0
310 FOR J%=1 TO 4
320 IF AN(J%)=AT(J%) THEN P=P+1:E$=E$+"+ ":AT(J%)=0
330 NEXT
340 FOR J%=1 TO 4
350 N=0
360 FOR K%=1 TO 4
370 IF J%=K% THEN 390
380 IF AT(J%)=AN(K%) THEN N=1
390 NEXT K%
400 IF N=1 THEN E$=E$+"- "
410 NEXT J%
420 PRINTTAB(10+17*(I%\8));E$
430 IF E$="+ + + + " THEN E=1
440 RETURN
450 FOR J%=1 TO 4
460 LOCATE (J%*2)-1,3:PRINTAN(J%)
470 NEXT
480 LOCATE 1,21
490 READ M$:IF E=1 THEN READ M$
500 PRINTM$
510 LOCATE1,22:PRINT"Press Space Bar To Play Again";
520 Z$=INPUT$(1):IF Z$<>" " THEN GOTO 510
530 RETURN
540 DATA "Hard luck - you ran out of goes","Well done"
```

Hangman

Summary

Although this might seem an essentially trivial problem to program, it proved quite tricky when I decided to use sprites for each letter of the alphabet in screen mode 2. Armed with 'Behind The Screens' I set about changing VDP registers, peeking and poking until a solution eventually popped out - definitely 90% perspiration, 10% inspiration! The method is shown below, and can easily be used in any program for moving hi-res text.

Also, if you wish to change the word list (the one I have included is pretty fiendish) you need only put the words in the data statements at line 360 onwards, and then change NW (number of words) in line 210 as appropriate.

When the program is run, there is a slight delay as the sprites are initialised, and then the display appears. You type the letter of your guess. If it is right the letter moves to the correct place and drops in. If it is wrong, it is added to a list at the top, and one more part of the gibbet or person is drawn.

Program Commentary

50-90	Main loop, calls subroutines to get letter, test it and see if won or lost.
100-130	Call win or lose routine as appropriate, end if needed.
140-200	Set up string array S$(0-25) as letters A-Z.
230	Load sprite data from S$.
240-290	Pick word randomly.
300-350	Set up display.
390-400	Clear letter input area.
460-490	Find position of guess in word if appropriate.
510-780	Move sprite letter to correct place. The two line TIME routines are delay loops - feel free to alter these to speed up or slow down the movement.
790-850	Routine to deal with wrong guess - place guess on 'wrong' pile, draw next part of hanging man.
860-890	Data for gibbet and man.
900-930	Win routine - play tune.
940-990	Lose routine - fill in correct word.

```
10 REM Hangman
20 CLEAR 500
30 GOSUB 140
40 GOSUB 210
50 GOSUB 390
60 GOSUB 460
70 IF N=L THEN W=1:GOTO 100
80 IF F=9 THEN W=0:GOTO 100
90 GOTO 50
100 IF W=1 THEN GOSUB 900 ELSE GOSUB 940
110 GOSUB 1000
120 IF AN$="Y" THEN RUN
130 END
140 SCREEN1
150 DIM S$(25)
160 FOR J=65 TO 90
170 FOR K=0 TO 7
180 S$(J-65)=S$(J-65)+CHR$(VPEEK(8*J+K))
190 NEXT:NEXT
200 RETURN
210 W=0:N=0:TR=0:NW=20:F=0
220 SCREEN 2:COLOR 15,4,7:OPEN "grp:" AS 1
230 FOR J=0 TO 25:SPRITE$(J)=S$(J):NEXT
240 WN=RND(-TIME)
250 WN=INT (RND(1)*NW)+1
260 RESTORE 360
270 FOR I=1 TO WN
280 READ W$
290 NEXT
300 L=LEN (W$):RESTORE 860
310 FOR I=10 TO 10*L STEP 10
320 DRAW"bm=I;,100 r8"
330 NEXT
340 PRESET (5,150):PRINT#1,"LETTER"
350 RETURN
360 DATA KVASS,BORZOI,ZYMURGY,POTATO,CONVINCE,DRIVEN,FINCHES
370 DATA FERRET,ENERGY,BUFFER,SAMURAI,CURVE,NEVER,COUGH
380 DATA QUIVER,HALVE,ITCH,JAW,COIN,JUNGLE
390 LINE (66,158)-(58,150),4,BF
400 PUTSPRITE SN,(0,209)
410 A$=INPUT$(1)
420 IF A$<"A" OR A$>"Z" THEN GOTO 410
430 SN=ASC(A$)-65
440 PRINT#1,A$:PUTSPRITE SN,(58,150),15
450 RETURN
460 IF INSTR(W$,A$)=0 THEN GOSUB 790
470 FOR I=1 TO L
480 IF MID$(W$,1,1)=A$ THEN GOSUB 510
490 NEXT
500 RETURN
510 N=N+1
520 FOR J=150 TO 134 STEP-4
530 PUTSPRITE SN,(58,J),15
540 TIME=0
```

```
550 IF TIME<10 THEN 550
560 NEXT
570 FOR J=58 TO 2 STEP-4
580 PUTSPRITE SN,(J,134)
590 TIME=0
600 IF TIME<10 THEN 600
610 NEXT
620 FOR J=134 TO 80 STEP -4
630 PUTSPRITE SN,(2,J)
640 TIME=0
650 IF TIME<10 THEN 650
660 NEXT
670 FOR J=12 TO 2+10*I STEP 2
680 PUTSPRITE SN,(J,80)
690 TIME=0
700 IF TIME<5 THEN 700
710 NEXT
720 FOR J=1 TO 5
730 PUTSPRITE SN,STEP(0,2)
740 TIME=0
750 IF TIME<25 THEN 750
760 NEXT
770 PRINT#1,A$:PLAY"L32N90"
780 RETURN
790 F=F+1
800 X=10*(F MOD 11):Y=50+10*(F\11)
810 PRESET(X,Y):PRINT#1,A$
820 READ D$
830 DRAW D$
840 PLAY"L2N10"
850 RETURN
860 DATA "BM120,150R70","BM155,150U100","BM155,70E20"
870 DATA"BM155,50R50","BM205,50D5"
880 DATA "BM202,55G6D6F6R6E6U6H6L6","BM205,73D25"
890 DATA "BM185,78R40","BM205,98NG15F15"
900 PRESET(10,165)
910 PRINT#1,"WELL DONE"
920 PLAY"O5L8E+FEFE4C+4DEDED2C+DCDC4E4B4O3E4R64E2"
930 RETURN
940 PRESET (10,165)
950 PRINT#1,"HARD LUCK"
960 FOR J=1 TO L
970 PRESET (J*10+2,90):PRINT#1,MID$(W$,J,1)
980 NEXT
990 RETURN
1000 PRESET (10,175)
1010 PRINT#1,"AGAIN (Y/N)"
1020 ANS$=INPUT$(1)
1030 IF ANS$<>"Y" AND ANS$<>"N" THEN GOTO 1020
1040 RETURN
```

Brickout

Summary

Another old classic, the idea of this game is to use the bouncing ball to destroy the bricks at the top of the screen. The ball bounces off the walls at one of two angles (just to add to the confusion) and you must prevent it going off the bottom of the screen with the bat, which is moved left and right with the keys "Z" and "X". When you have destroyed all but five of the bricks a new screen is formed, and you gain an extra life. This keeps on until you are dead - i.e. you can't win, but it's great fun trying.

The first problem in writing this program was selecting a suitable screen mode. I chose mode 1 because I could get by with block graphics, and this mode allowed sprites (the bat and ball are sprites). This means that all the graphics can be VPOKEd to the screen, and information can be VPEEKed, given the correct base address. The base addresses can be found in Behind The Screens Of The MSX by Mike Shaw (published by Kuma Computers Ltd) where the whole process of accessing the screen is explained.

Program Commentary

40-50	Sets sprite trapping for bat and ball.
90-200	Main loop for controlling game, calling subroutines to do most of the work.
210-220	Call the final score routine, end the program.
230-250	Test for bat striking ball, reverse ball if needed.
280-320	Define character for bricks.
330	Set brick colour.
400-470	Draw bricks and walls.
570-640	This is the control routine for the ball striking the bricks. It has to decide when to reverse direction etc., and also if a wall is struck.
650-760	Move bat.
770-800	Clear brick, increment score, test for end of screen.
810-830	Set bounce angle for wall.
840-860	Bounce off top of screen.
910-970	Final score, play again routine.

```
10 REM brickout
20 QQ=RND(-TIME)
30 KEYOFF
40 ON SPRITE GOSUB 230
50 SPRITE ON
60 GOSUB 260:REM set up
70 GOSUB 400
80 GOSUB 650
90 GOSUB 540
100 GOSUB 570
110 IF C=1 THEN BY%=-BY%:GOSUB 770
120 IF C=3 THEN GOSUB 770:C=1
130 IF W=1 THEN BX%=-BX%:GOSUB 810
140 GOSUB 840
150 GOSUB 650
160 IF YB%=209 THEN LI=LI-1:IF LI=0 THEN GOTO 210 ELSE GOSUB 870:GOTO 90
170 GOSUB 650
180 LOCATE 6,0:PRINT"lives";LI;"score ";SC
190 IF TIME>100 THEN SPRITE ON
200 GOTO 90
210 GOSUB 910
220 END
230 IF ABS(YB%-DY%)<=4 AND (DX%-XB%<=4 OR XB%-DX%<=20) THEN BY%=-BY%
240 SPRITE OFF:TIME=0
250 RETURN
260 SCREEN1,0:COLOR 14,15
270 SC=0:H=0:LI=5:B=0:W=0:BX%=8:BY%=-8
280 FOR I%=1 TO 8
290 READ J%
300 VPOKE 1280+I%,J%
310 NEXT
320 DATA 0,64,64,64,64,64,64,0
330 VPOKE &H2014,32
340 S$=CHR$(24)+CHR$(60)+CHR$(126)+CHR$(255)+CHR$(126)+CHR$(60)+CHR$(24)+CHR$(0)
350 SPRITE$(1)=S$
360 SPRITE$(2)=STRING$(7,0)+CHR$(255)
370 SPRITE$(3)=STRING$(7,0)+CHR$(255)
380 DX%=56:DY%=168:D1%=56:D2%=64
390 RETURN
400 FOR I%=7 TO 26
410 FOR J%=4 TO 8
420 VPOKE (6144+I%+(J%*32)),160
430 NEXT:NEXT
440 FOR I%=0 TO 23
450 VPOKE(6150+(I%*32)),219
460 VPOKE(6170+(I%*32)),219
470 NEXT
480 PUTSPRITE1,(BX%,BY%),10
490 PUTSPRITE2,(DX%,DY%),3
500 PUTSPRITE3,(DX%+8,DY%),3
510 BB=0:YB%=161:BY%=-8
520 XB%=INT((RND(1)*4)+1)*8+56
530 RETURN
540 XB%=XB%+BX%:TX%=XB%\8
```

```
550 YB%=YB%+BY%:TY%=YB%\8
560 RETURN
570 W=0
580 IF C=1 AND SGN(BY%)=1 THEN C=2 ELSE C=0
590 T=VPEEK(6144+TX%+(TY%*32))
600 Tl=VPEEK(6144+TX%+(TY%*32)+SGN(BX%))
610 IF Tl=219 THEN W=1
620 IF T=160 AND C=0 THEN C=1
630 IF T=160 AND C=2 THEN C=3
640 RETURN
650 AN$=INKEY$
660 IF AN$="z" OR AN$="Z" THEN DX%=DX%-8:GOSUB 690
670 IF AN$="x" OR AN$="X" THEN DX%=DX%+8:GOSUB 730
680 RETURN
690 IF D1%<D2% THEN D2%=D2%-16 ELSE D1%=D1%-16
700 PUTSPRITE 2,(D1%,DY%)
710 PUTSPRITE 3,(D2%,DY%)
720 RETURN
730 IF D1%>D2% THEN D2%=D2%+16 ELSE D1%=D1%+16
740 PUTSPRITE 2,(D1%,DY%)
750 PUTSPRITE 3,(D2%,DY%)
760 RETURN
770 VPOKE(6144+TX%+(TY%*32)),32
780 SI=9-TY%:SC=SC+SI:BB=BB+1
790 IF BB=90 THEN LI=LI+1:GOTO 70
800 RETURN
810 IF ABS(BX%)=4 THEN BX%=8*SGN(BX%)
820 IF RND(1)<.1 THEN BX%=4*SGN(BX%)
830 RETURN
840 IF YB%<=16 THEN BY%=-BY%
850 PUTSPRITE1,(XB%,YB%)
860 RETURN
870 XB%=(INT(RND(1)*4+1)*8)+56
880 YB%=161:BX%=8:BY%=-8
890 FOR T=1 TO 500:NEXT T
900 RETURN
910 FOR I=1 TO 1000:NEXT
920 SCREEN 0
930 LOCATE 16,10
940 PRINT"score ";SC
950 LOCATE 16,14
960 PRINT"Again (Y/N)"
970 AN$=INPUT$(1)
980 IF AN$="Y" OR AN$="y" THEN RUN
990 IF AN$="N" OR AN$="n" THEN RETURN
1000 GOTO 970
```

Blast The Mutant Gerbils

Summary

At last, a game of the 'boot the wombat' variety. Mindless, childish, inane, but great fun! In this game you are the last hope of saving a small isolated village from the clutches of the incoming gerbil spaceships. Using either a joystick, or keyboard, you must fire missiles at the approaching nasties and destroy them, the object being (apart from enjoying yourself) to to kill as many as possible before the aliens reach the houses at the bottom of the screen. Your supply of missiles is unlimited, but you have a random supply of hypermissiles which destroy a correctly targeted ship immediately on launch.

The program is written so as to demonstrate some of the more exotic features of the MSX, namely sprite, interval, function key and joystick fire button trapping. The game is much easier when played with a joystick. Keyboard control uses the function key trap but could be replaced by a more conventional INKEY$ type routine to use more convenient keys.

Program Commentary

100-170   Sets up screen and variables and asks for instructions.

180-200   Clears keyboard buffer, checks for valid input and prints instructions if requested.

210-240   Selects hi-res screen with magnified sprites and no key prompts and sets a string to record the existance of alien sprites.

250-310   Sets alien coordinates, defines sprites and draws houses.

320-380   Set up function key, interval, sprite and fire button trapping.

390-510   Main Program Loop :

400   Fires missile if fire button or space bar pressed by enabling interval trapping, moving missile up every 0.04s.

420-450   Move alien sprites (U%=Level no, Q%=Movement step per frame).

460   Check for missile hitting alien, blank & disable if hit.

470-480   If near screen edges bounce aliens off and move Q% pixels down every 0.4s.

490-510   Move sprites and check for no aliens left.

520-530   Disable all trapping and if all aliens destroyed
              increment level and re-run,else print "You're
              dead" etc.

580-610,
780-800,
870-920   Define alien, gun and missile sprites.

720-740   Draw four houses using routine in lines 620-710.

750-770   Routine called by interval trapping to move missile up.

810-860   Set up missile coordinates, enable interval trapping
          and stop subsequent missile firing.

930-1010 "You're dead,Play again ?" routine.

1040-1090 Routine called by sprite trapping. Finds which sprites
          have collided and takes necessary action.

1100-1360 Print instructions.

1370-1380,
1390-1400 Move gun left/right.

1460-1530 Separate subroutines called to blank aliens when hit.

    Note that when a missile passes between two alien ships, the
rightmost one is blanked. This is because the MSX can only handle
four sprites on the same line, the highest numbered one being
blanked by the VDP chip while on the same line as the other four.

```
10 REM ******************************
20 REM **                        **
30 REM ** ´BLAST THE MUTANT GERBILS´**
40 REM **         Written          **
50 REM **    7th January 1985.     **
60 REM **                        **
70 REM ******************************
80 REM
90 REM
100 ONERRORGOTO1420
110 OPEN"GRP:"AS#1
120 COLOR15,0,0
130 KEYOFF
140 DIMS$(4)
150 U%=1
160 TIME=0
170 SCREEN0,3,0:P$="Do you want instructions ?":P%=4:GOSUB1020
180 FORA=0TO30:I$=INKEY$:NEXT
190 I$=INKEY$:IFI$<>"Y"ANDI$<>"y"ANDI$<>"N"ANDI$<>"n"THEN190
200 IFI$="Y"ORI$="y"THENGOSUB1110
210 FORA=0TO4:S$(A)="A":NEXT
220 SCREEN2,3,0
230 KEYOFF
240 Q%=1
250 REM ** INITIALISE X COORDS FOR ALIENS `**
260 A%=25:B%=81:C%=137:D%=193:K%=10:MF%=0
270 GOSUB730
280 GOSUB590
290 P$="LEVEL "+STR$(U%):PRESET(250-8*LEN(P$),2):PRINT#1,P$
300 W%=128:N%=128
310 GOSUB780:GOSUB880
320 DRAW"BM128,175"
330 STRIG(0)ON
340 PUTSPRITE0,(128,175),15
350 ONSTRIGGOSUB820
360 SPRITEON:ONSPRITEGOSUB1050
370 KEY(1)ON:KEY(2)ON:ONKEYGOSUB1370,1390
380 ONINTERVAL=2GOSUB760:REM MOVE MISSILE EVERY 2 CLOCK CYCLES (IF FIRED)
390 I%=STICK(1)
400 IFSTRIG(1)ANDNOTMF%THENGOSUB820
410 IFTIMEMOD20<5THENPLAY"L64O1C"
420 A%=(A%+(U%*Q%))MOD256:PUTSPRITE2,(A%,K%),10
430 B%=(B%+(U%*Q%))MOD256:PUTSPRITE3,(B%,K%),2
440 C%=(C%+(U%*Q%))MOD256:PUTSPRITE4,(C%,K%),5
450 D%=(D%+(U%*Q%))MOD256:PUTSPRITE5,(D%,K%),8
460 IFABS(F%-K%)<20ANDABS(D%-W%)<20THENGOSUB1460
470 IFA%<7ORD%>240THENQ%=-Q%
480 IFTIMEMOD20<10THENK%=K%+U%
490 IFI%=7ANDN%>0THENGOSUB1470ELSEIFI%=3ANDN%<240THENN%=N%+4:PUTSPRITE0,(N%,175),15
500 T$=S$(0)+S$(1)+S$(2)+S$(3)
510 IFK%<129ANDT$<>""THEN390
520 INTERVALOFF:STRIG(0)OFF:STRIG(1)OFF:KEY(1)OFF:KEY(2)OFF:SPRITEOFF
530 IFT$=""THENU%=U%+1:PLAY"O3L8CCCDCDEL64":GOTO210ELSEGOSUB930
540 FORA=0TO45:I$=INKEY$:NEXT
```

```
550 I$=INKEY$:IFI$<>" "ANDI$<>"E"THEN550
560 IFI$=" "THENRUN
570 SCREEN0:END
580 REM ** DEFINE GUN SPRITE **
590 S$=STRING$(3,CHR$(24))+"<~~~"+CHR$(255)
600 SPRITE$(0)=S$
610 RETURN
620 REM ** HOUSE ROUTINE **
630 PSET(V%,170),11:REM LEFT HAND EDGE =V%
640 H%=V%+10:DRAW"U20E5R9U3L1R3C15U1L1U2L1U1R1L1D1R1D2R1D1R2C11L1D3R5F5D20"
650 DRAW"L26BM=H%;,170U10R6D10"
660 W$="R6U4L6D4"
670 FORO%=V%+2TOV%+18STEP16
680 FORH%=159TO167STEP8
690 DRAW"BM=O%;,=H%;XW$;"
700 NEXT:NEXT
710 RETURN
720 REM ** DRAW HOUSES **
730 FORV%=25TO200STEP56:GOSUB630:PAINT(V%+5,160),11:NEXT
740 RETURN
750 REM ** MOVE MISSILE UP **
760 IFF%>-32THENGOSUB1520ELSEINTERVALOFF:STRIG(0)ON:MF%=0
770 RETURN
780 REM ** DEFINE MISSILE SPRITE **
790 SPRITE$(1)=STRING$(8,CHR$(24))
800 RETURN
810 REM ** FIRE ROUTINE **
820 W%=N%:F%=159:MF%=-1
830 PLAY"O3C"
840 INTERVALON
850 STRIG(0)OFF
860 RETURN
870 REM ** DEFINE ALIEN SPRITES **
880 SPRITE$(2)=STRING$(2,CHR$(153))+"Z<B$B"+CHR$(129)
890 SPRITE$(3)=CHR$(68)+"|"+CHR$(16)+CHR$(40)+CHR$(84)+CHR$(40)+STRING$(2,CHR$(16))
900 SPRITE$(4)=STRING$(2,CHR$(16))+"8|"+CHR$(254)+"8"+CHR$(16)
910 SPRITE$(5)="AAA]JA"+CHR$(127)+CHR$(8)
920 RETURN
930 SCREEN0
940 P$="You´re People Are Now Themselves":P%=4:GOSUB1020
950 P$="Mutant Gerbils - You Have Failed":P%=6:GOSUB1020
960 P$="Your Brave Last Stand Lasted":P%=10:GOSUB1020
970 P$=STR$(INT(TIME/50))+" Seconds  ":P%=12:GOSUB1020
980 P$="Ending Tragically on Level "+STR$(U%):P%=14:GOSUB1020
990 P$="If you wish to try again press SPACE":P%=16:GOSUB1020
1000 P$="Otherwise press E":P%=18:GOSUB1020
1010 RETURN
1020 LOCATE19-LEN(P$)/2,P%:PRINTP$;
1030 RETURN
1040 REM ** TRAP SPRITE COLLISIONS **
1050 BEEP:IFABS(W%-A%)<25ANDS$(0)<>""THENGOSUB1480:RETURN
1060 IFABS(W%-B%)<25ANDS$(1)<>""THENGOSUB1490:RETURN
1070 IFABS(W%-C%)<25ANDS$(2)<>""THENGOSUB1500:RETURN
1080 IFABS(W%-D%)<25ANDS$(3)<>""THENGOSUB1510:RETURN
```

```
1090 RETURN
1100 REM ** INSTRUCTIONS **
1110 P$="Your tranquil home town is the scene":P%=0:GOSUB1020
1120 P$="Of a planetary invasion from Mars.":P%=1:GOSUB1020
1130 P$="The only hope of stopping the invading":P%=2:GOSUB1020
1140 P$="Mutant Gerbils is to destroy their":P%=3:GOSUB1020
1150 P$="Attack waves before they can land":P%=4:GOSUB1020
1160 P$="And take the village's inhabitants":P%=5:GOSUB1020
1170 P$="Hostage and subject them to horrible,":P%=6:GOSUB1020
1180 P$="Torturing experiments as they try to":P%=7:GOSUB1020
1190 P$="Perfect their latest secret weapon-":P%=8:GOSUB1020
1200 P$="A new type of nerve gas which,if":P%=9:GOSUB1020
1210 P$="Released would wipe out the human":P%=10:GOSUB1020
1220 P$="Race in readiness for the planet":P%=11:GOSUB1020
1230 P$="To be colonised by 10 foot tall,green":P%=12:GOSUB1020
1240 P$="Gerbils.":P%=13:GOSUB1020
1250 P$="Blast the incoming ships before":P%=14:GOSUB1020
1260 P$="They can reach the peaceful village":P%=15:GOSUB1020
1270 P$="Below and carry out their plan.":P%=16:GOSUB1020
1280 P$="Manoeuvre yourself directly under the":P%=18:GOSUB1020
1290 P$="Aliens using F1 for left & F2 for right":P%=19:GOSUB1020
1300 P$="And fire using the SPACE BAR":P%=20:GOSUB1020
1310 P$="Good shooting and good luck!":P%=21:GOSUB1020
1320 P$="PRESS SPACE BAR TO START":P%=23:GOSUB1020
1330 REM EMPTY KEYBOARD BUFFER
1340 FORA=0TO30:I$=INKEY$:NEXT
1350 IFINKEY$<>" "THEN1350
1360 RETURN
1370 IFN%>0THENN%=N%-4:PUTSPRITE0,(N%,175),15
1380 RETURN
1390 IFN%<240THENN%=N%+4:PUTSPRITE0,(N%,175),15
1400 RETURN
1410 REM ** TIME OUT TRAP **
1420 IFERR=6THENTIME=0:RUN
1430 SCREEN0:KEYON
1440 PRINT"Program error ";ERR;" at line ";ERL
1450 STOP
1460 SPRITE$(5)="":S$(3)="":F%=-32:PUTSPRITE1,(W%,-32),15:BEEP:RETURN
1470 N%=N%-4:PUTSPRITE0,(N%,175),15
1480 SPRITE$(2)="":S$(0)="":F%=-32:PUTSPRITE1,(W%,-32),15:RETURN
1490 SPRITE$(3)="":S$(1)="":F%=-32:PUTSPRITE1,(W%,-32),15:RETURN
1500 SPRITE$(4)="":S$(2)="":F%=-32:PUTSPRITE1,(W%,-32),15:RETURN
1510 SPRITE$(5)="":S$(3)="":F%=-32:PUTSPRITE1,(W%,-32),15:RETURN
1520 F%=F%-2:PUTSPRITE1,(W%,F%),15
1530 RETURN
```

Summary

The object of the game is to eat all the red pearls in the maze whilst avoiding the ghosts who haunt it. There are three ghosts, each with distinct personalities.

Happy : A simple ghost, content with his death who loves wandering around the maze.

Grumpy : He suffers from NBI (Not Bloomin' Interested) and stomps around in the smallest circle he has found in the maze in total dejection.

Frumpy : He is by turns enthusiastic and uninterested. He sometimes races other ghosts out of their lair, yet floats over the top of your man without killing him justs moments later - watch out though as he can be deadly.

To be touched by a ghost is death (with the occasional exception of Frumpy), unless you have eaten one of the blue power storing gems in the corners of the maze. This gives you the power to destroy the ghosts for a while, turning them white with terror. When you destroy a ghost it returns directly to its home and cannot then harm you until the final burst of energy from the gem causes it (the ghost) to go blue.

Once you have cleared the maze of the red pearls, you are transported to another identical maze where success is not so easy.

The program uses some clever graphics - the most obvious being the fact that different characters have different colours when screen 1 only allows one foreground colour. In this mode the colour of a character is defined at VRAM location (8192 + ASC(CHAR)/8) and can be set by VPOKE. You poke an eight bit binary number to the location - the first four bits define the foreground colour number, the second four the background. Alternatively VPOKE the number (16 x fore) + back (decimal). This number also affects seven other characters, as the colours are stored for blocks of eight characters starting from 32.

E.g. changing the colour of 'O' also changes the seven characters before it, as they share a common block of eight VRAM locations.

Program Commentary

Line	Description
10-300	Display maze on screen.
310-350	Redefine shape of 'O' and 'P'.
360-380	Redefine shape of '.'.
390-400	Change colour of '.' and 'P'.
410-730	Define sprite shapes.
740-830	Display sprites, set locations.
840-870	Test to see if found gems or pearls.
880-900	Takes appropriate action if gem found, changes level if needed.
910-990	Tests to see if man should be moved.
1000-1050	Moves 'Happy' ghost.
1060-1110	Moves 'Frumpy' ghost.
1120-1180	Moves 'Grumpy' ghost.
1190-1220	Sprite data for man.
1230-1330	Sprite data for ghosts.
1340-1460	Ghost movement data.
1470-1580	Tests for collisions with the wall of the maze.
1590-1610	Next level - pause, change level number.
1620-1660	Routine for when power gem is found.
1670-1800	Tests for collision between man and ghosts.
1810-1880	Routine for when life lost.
1890-1970	End game and rerun.
1980-2030	Subroutines called from power gem routine.

```
10 KEYOFF
20 CLEAR200
30 SCREEN1,2,0
40 SPRITEON
50 ONSPRITEGOSUB1670
60 COLOR7,0,0
70 LV%=3:L%=1
80 PRINT"OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO"
90 PRINT"OP.......... OO...........P O"
100 PRINT"O.        .   . OO,    .      . O"
110 PRINT"O. OOOO. OO..... OO. OOOO. O"
120 PRINT"O. OOOO. OO.    . OO. OOOO. O"
130 PRINT"O........................ O"
140 PRINT"O  .   .        .    . O"
150 PRINT"OOO. OO. OOOO  OOOO. OO. OOO"
160 PRINT"OOO. OO. O  O   O  O. OO. OOO"
170 PRINT".... OO. O        O. OO....."
180 PRINT"   . OO. O        O. OO."
190 PRINT"OOO. OO. O        O. OO. OOO"
200 PRINT"OOO. OO. OOOOOOOOOO. OO. OOO"
210 PRINT"O................ ............ O"
220 PRINT"O.       .        .     . O"
230 PRINT"O. OOOO. OOOO. OOOO. OOOO. O"
240 PRINT"O. OOOO. OOOO. OOOO. OOOO. O"
250 PRINT"O. OOOO... OO. OO... OOOO. O"
260 PRINT"O. OOOO... OO. OO... OOOO. O"
270 PRINT"OP......................P O"
280 PRINT"O                         O"
290 PRINT"OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO"
300 IFL%>1THENPRINT
310 FORX%=0TO7
320 VPOKE(79*8)+X%,255
330 VPOKE(80*8)+X%,127
340 NEXT
350 VPOKE(80*8),0
360 VPOKE(46*8)+6,3
370 VPOKE(46*8)+7,3
380 VPOKE(46*8)+5,0
390 VPOKE(46/8)+8192,&H60
400 VPOKE(80/8)+8192,&H40
410 IFL%>1THENGOTO700
420 RESTORE1190
430 FORX%=1TO32
440 READA%
450 SP$=SP$+CHR$(A%)
460 NEXT
470 FORX%=1TO32
480 READA%
490 G1$=G1$+CHR$(A%)
500 NEXT
510 FORX%=1TO32
520 READA%
530 G2$=G2$+CHR$(A%)
540 NEXT
```

```
550 FORX%=1TO32
560 READA%
570 G3$=G3$+CHR$(A%)
580 NEXT
590 DIMHX%(15):DIMHY%(15):DIMGX%(30)
600 DIMFX%(7):DIMFY%(7):DIMGY%(30)
610 FORX%=1TO15
620 READHX%(X%),HY%(X%)
630 NEXT
640 FORX%=1TO30
650 READGX%(X%),GY%(X%)
660 NEXT
670 FORX%=1TO7
680 READFX%(X%),FY%(X%)
690 NEXT
700 SPRITE$(1)=SP$
710 SPRITE$(2)=G1$
720 SPRITE$(3)=G2$
730 SPRITE$(4)=G3$
740 X1%=119:X2%=103:X3%=117:X4%=133
750 Y1%=103:Y2%=75:Y3%=75:Y4%=75
760 C%=119:D%=103:H%=1:G%=30:F%=7
770 I%=11:J%=13:M%=12
780 IFC%=>239THENC%=7
790 IFC%=<-1THENC%=239
800 PUTSPRITE1,(C%,D%),9,1
810 PUTSPRITE2,(X2%,Y2%),I%,2
820 PUTSPRITE3,(X3%,Y3%),J%,3
830 PUTSPRITE4,(X4%,Y4%),M%,4
840 SPRITEON
850 B%=VPEEK(6177+INT(C%/8)+(INT(D%/8)*32))
860 IFB%=46THENVPOKE(6177+INT(C%/8)+(INT(D%/8)*32)),32:BEEP:SP%=SP%+100
870 IFB%=80THENGOTO1620
880 P%=P%-1:IFP%=0THENGOTO770
890 IFP%>0ANDP%<10THENI%=5:J%=5:M%=5
900 IFSP%=18000*L%THENGOTO1590
910 K$=INKEY$
920 IFK$=""THENGOTO 1000
930 K%=ASC(K$)
940 IFK%=31THENGOTO1470
950 IFK%=30THENGOTO1500
960 IFK%=28THENGOTO1530
970 IFK%=29THENGOTO1560
980 IFC%=>239THENC%=7
990 IFC%=<7THENC%=239
1000 IFX2%<HX%(H%)THENX2%=X2%+2
1010 IFX2%>HX%(H%)THENX2%=X2%-2
1020 IFY2%<HY%(H%)THENY2%=Y2%+2
1030 IFY2%>HY%(H%)THENY2%=Y2%-2
1040 IFX2%=HX%(H%)ANDY2%=HY%(H%)THENH%=H%+1
1050 IFH%=16THENH%=1
1060 IFX4%<FX%(F%)THENX4%=X4%+2
1070 IFX4%>FX%(F%)THENX4%=X4%-2
1080 IFY4%<FY%(F%)THENY4%=Y4%+2
```

```
1090 IFY4%>FY%(F%)THENY4%=Y4%-2
1100 IFX4%=FX%(F%)ANDY4%=FY%(F%)THENF%=F%-1
1110 IFF%=0THENF%=7
1120 IFX3%<GX%(G%)THENX3%=X3%+2
1130 IFX3%>GX%(G%)THENX3%=X3%-2
1140 IFY3%<GY%(G%)THENY3%=Y3%+2
1150 IFY3%>GY%(G%)THENY3%=Y3%-2
1160 IFX3%=GX%(G%)ANDY3%=GY%(G%)THENG%=G%-1
1170 IFG%=0THENG%=30
1180 GOTO780
1190 DATA7,31,57,57,127,127,0,0,3,15,127
1200 DATA63,31,15,3,0,192,240,248,248,252
1210 DATA252,124,252,252,252,248,248,240
1220 DATA224,128,0
1230 DATA0,1,3,15,12,8,15,15,13,14,15,15
1240 DATA15,12,0,0,0,240,248,254,230,66
1250 DATA254,254,246,238,30,254,254,204
1260 DATA0,0,0,1,3,15,12,8,15,15,12,15,15
1270 DATA15,15,12,0,0,0,240,248,254,230
1280 DATA66,254,254,254,30,230,254,254
1290 DATA204,0,0
1300 DATA0,1,3,15,12,8,15,15,15
1310 DATA14,13,15,15,12,0,0,0,240,248,254
1320 DATA230,66,254,254,30,238,246
1330 DATA254,254,204,0,0
1340 DATA117,75,117,37,23,37,23,7,71,7
1350 DATA71,151,23,151,23,103,215,103,215,151
1360 DATA167,151,167,7,215,7,215,37
1370 DATA117,37
1380 DATA117,75,117,37,103,23,103,7,71
1390 DATA7,71,39,39,39,39,71,11,71
1400 DATA39,71,39,103,71,103,71,137,87,151
1410 DATA117,151,117,103,117,151,153
1420 DATA151,165,135,165,103,199,103
1430 DATA199,71,255,71,199,71,199,39
1440 DATA167,39,167,7,135,7,135,23,117,37
1450 DATA117,75,117,37,71,37,71,103
1460 DATA165,103,165,37,117,37
1470 V%=VPEEK(6209+INT(C%/8)+(INT(D%/8)*32))
1480 IFV%<>79THEND%=D%+5-L%:IF5-L%<1THEND%=D%+1
1490 GOTO 1000
1500 V%=VPEEK(6145+INT(C%/8)+(INT(D%/8)*32))
1510 IFV%<>79THEND%=D%-5+L:IF5-L%<1THEND%=D%-1
1520 GOTO 1000
1530 V%=VPEEK(6178+INT(C%/8)+(INT(D%/8)*32))
1540 IFV%<>79THENC%=C%+5-L%:IF5-L%<1THENC%=C%+1
1550 GOTO 1000
1560 V%=VPEEK(6176+INT(C%/8)+(INT(D%/8)*32))
1570 IFV%<>79THENC%=C%-5+L%:IF5-L%<1THENC%=C%-1
1580 GOTO 1000
1590 L%=L%+1:SCREEN1
1600 LOCATE10,15:PRINT"LEVEL";L%
1610 FORX=1TO1000:NEXT:GOTO80
1620 VPOKE(6177+INT(C%/8)+(INT(D%/8)*32)),32
```

```
1630 P%=INT(RND(1)*200)
1640 PLAY"T255ABDEF"
1650 I%=15:J%=15:M%=15
1660 GOTO900
1670 SPRITEOFF
1680 IFX2%+14=>C%ANDX2%=<C%+14THENGOTO1720
1690 IFX3%+14=>C%ANDX3%=<C%+14THENGOTO1720
1700 IFX4%+14=>C%ANDX4%=<C%+14THENGOTO1720
1710 RETURN
1720 IFY2%+14=>D%ANDY2%=<D%+14THENGOTO1760
1730 IFY3%+14=D%ANDY3%=<D%+14THENGOTO1760
1740 IFY4%+14=>D%ANDY4%=<D%+14THENGOTO1760
1750 RETURN
1760 IFP%<1THENGOTO1810
1770 IFX2%+14=>C%ANDX2%=<C%+14ANDY2%=>D%ANDY2%=<D%+14ANDI%=15THENGOSUB1980
1780 IFX3%+14=>C%ANDX3%=<C%+14ANDY3%+14=>D%ANDY3%=<D%+14ANDJ%=15THENGOSUB2000
1790 IFX4%+14=>C%ANDX4%=<C%+14ANDY4%+14=>D%ANDY4%=<D%+14ANDM%=15THENGOSUB2020
1800 RETURN
1810 LOCATE11,10
1820 LV%=LV%-1
1830 PRINT"LIFE";LV%;
1840 FORX=1TO1000:NEXTX
1850 LOCATE11,10
1860 PRINT"        ";
1870 IFLV%=0THENGOTO1890
1880 GOTO740
1890 SCREEN0
1900 LOCATE10,10
1910 PRINT"YOU SCORED";SP%+SG%
1920 LOCATE10,15
1930 PRINT"ANOTHER GAME (Y/N)"
1940 K$=INKEY$
1950 IFK$="Y"THENRUN
1960 IFK$="N"THENEND
1970 GOTO1940
1980 H%=1:SG%=SG%+500:I%=11:PLAY"T255FEDBA"
1990 RETURN
2000 G%=30:SG%=SG%+500:J%=13:PLAY"T255FEDBA"
2010 RETURN
2020 P%=7:SG%=SG%+500:M%=12:PLAY"T255FEDBA"
2030 RETURN
```

Lunar Lander

Summary

The Eagle has landed ! Or will do if your skill as a pilot can safely guide the LEM (lunar excursion module) to the ground. Using horizontal and vertical thrusters you have to position the module on the landing pad (supplied by friendly moon-people presumably) in a nice controlled descent under gravity. If you pass the first level, the gravitational pull increases and you start with less fuel. Your score depends only on your dying effort and the number of levels you pass - so button up your space suit, and don't forget to close the airlock !

Program Commentary

10-250   Instructions, initiate variables.

260-380   Draw screen, including fuel and speed meters.

390-430   Define sprite for LEM.

440-480   Draw landing pad and LEM.

490   Increase vertical velocity due to gravity.

500-630   Input move instruction - check for edge of screen.

640-770   Update fuel and velocity graphs.

780-810   Check landing - redirect control accordingly.

820-1020   Successful landing - go to next level.

1030-1230   Crash landing - death and final score.

1240-1270   Sprite data.

1280-1350   Set graphics characters.

```
10 COLOR 15,0,0
20 SCREEN0:KEYOFF
30 OPEN"GRP:"AS#1
40 PRINT" LL      AAA  N   N DD     EEEEE RRR"
50 PRINT" LL      A  A NN  N D D    E      R  R"
60 PRINT" LL      AAAAA N N N D  D  EEE    RRRR"
70 PRINT" LL      A  A N  NN D D    E      R R"
80 PRINT" LLLLL A   A N   N DD      EEEEE R  R"
90 PRINT:PRINT:PRINT
100 PRINT"  The object of this game is to"
110 PRINT
120 PRINT" land the space craft on the "
130 PRINT
140 PRINT" landing pad .The craft will abort"
150 PRINT
160 PRINT" ,killing all occupants ,including"
170 PRINT
180 PRINT" you !!!!!!!!!!!!"
190 L=1:LVL=1:F=1000
200 Y=20:X=RND(1)*256
210 LOCATE5,20
220 PRINT"PRESS ANY KEY TO CONTINUE"
230 K$=INKEY$
240 IFK$<>""THENGOTO260
250 GOTO230
260 SCREEN2,2
270 FORX=1TO50
280 COLORRND(1)*13,1,1
290 PSET(RND(1)*256,RND(1)*140)
300 NEXT
310 COLOR15,0,0
320 LINE(0,140)-(256,140),15
330 PRESET(0,155)
340 PRINT#1,"HORIZONTAL VELOCITY"
350 PRESET(0,175)
360 PRINT#1,"VERTICAL VELOCITY"
370 PRESET(0,185)
380 PRINT#1,"FUEL"
390 FORX=0TO31
400 READA
410 SP$=SP$+CHR$(A)
420 NEXT
430 SPRITE$(1)=SP$
440 BX=INT(RND(1)*230)
450 FORBY=141TO145
460 LINE(BX,BY)-(BX+30,BY),15
470 NEXT
480 PUTSPRITE1,(X,Y),7,1
490 VVECT=VVECT+L/8
500 K$=INKEY$
510 I=I+1
520 IFK$=""THENGOTO580
530 K=ASC(K$)
540 IFK=28THENHVECT=HVECT+2:F=F-10
```

```
540 IFK=28THENHVECT=HVECT+2:F=F-10
550 IFK=29THENHVECT=HVECT-2:F=F-10
560 IFK=30THENVVECT=VVECT-3:F=F-30
570 IFK=31THENVVECT=VVECT+3:F=F-30
580 X=X+HVECT:Y=Y+VVECT
590 IFX<=0THENX=0
600 IFX>=235THENX=235
610 IFY<0THENY=0
620 IFF<=0THENGOTO1030
630 IFY>=123THENGOTO780
640 COLOR0,0,0
650 LINE(20,150)-(255,153),,BF
660 COLOR15+12*(HVECT>0),0,0
670 LINE(20,150)-(ABS(10*HVECT)+20,153),,BF
680 COLOR0,0,0
690 LINE(20,170)-(255,173),,BF
700 COLOR15+12*(VVECT>0),0,0
710 LINE(20,170)-(ABS(10*VVECT)+20,173),,BF
720 COLOR0,0,0
730 LINE(40,190)-(255,193),,BF
740 COLOR7,0,0
750 LINE(40,190)-((F/10)+40,193),,BF
760 COLOR7,0,0
770 GOTO480
780 IFX=>BXANDX<=BX+15THENGOTO800
790 GOTO1030
800 IFHVECT<=0ANDVVECT<=3THENGOTO820
810 GOTO1030
820 SCREEN0
830 PRINT"      W   W EEEEE LL      LL"
840 PRINT"      W   W E     LL      LL"
850 PRINT"      W W W EEEE  LL      LL"
860 PRINT"      W W W E     LL      LL"
870 PRINT"       W W  EEEEE LLLLL LLLLL"
880 PRINT:PRINT:PRINT
890 PRINT"       DD     OOO  N   N EEEEE"
900 PRINT"       D D   O   O NN  N E"
910 PRINT"       D  D  O   O N N N EEEE"
920 PRINT"       D D   O   O N  NN E"
930 PRINT"       DD     OOO  N   N EEEEE"
940 PLAY"T255ACDEFGCDEAF"
950 PRINT:PRINT:PRINT:PRINT
960 PRINT"        NOW FOR LEVEL"LVL+1
970 LVL=LVL+1:
980 L=L+1.5:I=0
990 IFL=6THENL=1:F=1000-(LVL/5*20)
1000 FORX=1TO1000:NEXT
1010 SP$="":RESTORE
1020 GOTO200
1030 CLS
1040 SCREEN0
1050 PRINT" Y   Y OOO  U   U ´´   RRR  EEEEE"
1060 PRINT"  Y Y O   O U   U  ´´  R  R E"
1070 PRINT"   Y  O   O U   U   ´  RRRR EEEE"
1080 PRINT"   Y  O   O U   U      R  R E"
```

```
1090 PRINT"    Y     OOO    UUU        R  R EEEEE"
1100 PRINT:PRINT:PRINT
1110 PRINT"         DD    EEEEE  AAA   DD    !!"
1120 PRINT"         D D   E      A   A D D   !!"
1130 PRINT"         D  D EEEE   A    A D  D !!"
1140 PRINT"         D D   E     AAAAA D D"
1150 PRINT"         DD    EEEEE A   A  DD    !!"
1160 PRINT:PRINT:PRINT
1170 PRINT"         YOU SCORED ";(F*10-I*10)*(LVL-1)
1180 PRINT:PRINT
1190 PRINT"            ANOTHER GAME ?"
1200 K$=INKEY$
1210 IFK$="Y"THENRUN
1220 IFK$="N"THENEND
1230 GOTO1200
1240 DATA 0,1,3,6,12,5,14,14,31,25
1250 DATA 17,17,17,57,41,124,0,192
1260 DATA 224,176,152,248,56,56,252
1270 DATA 76,196,68,192,70,202,31
1280 RESTORE1350
1290 FORY=1TO6
1300 READP%
1310 FORX=0TO7
1320 VPOKE(2048+(P*8)),255
1330 NEXT
1340 RETURN
1350 DATA 76,65,78,44,69,82
```

Maze

Summary

Enjoy tactical games ? Prefer fast-action arcade games ? This program combines the best of both, the idea being to trap a little bouncing ball with your tail. In doing so however there is every possibility of trapping yourself ! To further increase the adrenalin flow, the clock ticks away at the top of the screen. Quick thinking is vital to avoid running yourself into a stalemate, and having to clear your tail, wasting valuable seconds.

Program Commentary

40-150   Sets up tail data, UDG characters.

380-440   Draws screen display.
a
450-470   Sets variables for real-time play.

480-580   Moves player if requested, and possible.

610-710   Directs randomly bouncing ball.

650-730   Moves ball, checks to see if trapped.

740-760   Clear tail if needed.

```
10 REM
20 REM   MAZE CHASE
30 REM
40 CLEAR 200
50 DIMA((40*22)-150)
60 SCREEN 0
70 KEYOFF
80 POKE&HF3DB,1
90 COLOR 1,4,7
100 CLS
110 RESTORE
120 FORR=0 TO 7:READS$:S=VAL("&B"+S$+"00")
130 VPOKE2048+(8*97)+R,S
140 NEXT
150 DATA000000,000000,000000,001100,001100,000000,000000,000000
160 PRINT
170 PRINT"    MAZE CHASE"
180 PRINT"    ----------"
190 PRINT
200 PRINT
210 PRINT" THE AIM OF THE GAME"
220 PRINT" IS TO TRAP THE ´o´"
230 PRINT" IN THE MAZE WITH YOUR ´*´"
240 PRINT
250 PRINT" TO STEER YOUR PLAYER USE"
260 PRINT" Z  X  :  /"
270 PRINT" TO MOVE LEFT,RIGHT,UP AND DOWN."
280 PRINT
290 PRINT" IF YOU GET TRAPPED PRESS RETURN      TO CLEAR YOUR TAIL"
300 PRINT
310 PRINT" TO ABORT PRESS [ESC]"
320 PRINT
330 PRINT
340 PRINT
350 PRINT" PRESS ANY KEY TO START"
360 A$=INKEY$
370 A$=INKEY$:IFA$=""THENGOTO370
380 CLS
390 FORR=1TO200
400 P=INT(RND(1)*(22*40))
410 IFVPEEK(P)=32THENVPOKEP,97
420 NEXT
430 FORR=0TO39:VPOKER,97:VPOKE(22*40)+R,97:NEXT
440 FORR=1 TO 21:VPOKE(R*40),97:VPOKE(R*40)+39,97:NEXT
450 X=INT(RND(1)*(22*40)):IFVPEEK(X)<>32THEN GOTO 450
460 M=INT(RND(1)*(22*40)):IFVPEEK(M)<>32THEN GOTO 460
470 TIME=0
480 A$=INKEY$
490 A$=INKEY$:IFA$=""THEN GOTO 590
500 IF(A$="Z"ORA$="z")ANDVPEEK(X-1)=32THEND=-1
510 IF(A$="X"ORA$="x")ANDVPEEK(X+1)=32THEND=+1
520 IF A$="/"ANDVPEEK(X+40)=32THEND=+40
530 IF A$=":"ANDVPEEK(X-40)=32THEND=-40
540 IF A$=CHR$(13)THEN GOSUB 720
```

```
550 IF A$=CHR$(27) THEN GOTO 20
560 X=X+D:VPOKEX,ASC("*")
570 D=0
580 A(Q)=X:Q=Q+1
590 LOCATE8,23:PRINT" TIME ;";INT(TIME/50);:D=0
600 PM=INT(RND(1)*4)+1
610 ONPMGOTO630,640,650,660
620 GOTO600
630 K=-1:GOTO670
640 K=1:GOTO670
650 K=40:GOTO670
660 K=-40
670 IFVPEEK(M+1)=32ORVPEEK(M-1)=32ORVPEEK(M+40)=32ORVPEEK(M-40)=32THENGOTO680ELSE750
680 IFVPEEK(M+K)<>32 THEN GOTO 490
690 IF VPEEK(M+K)=32 THEN VPOKEM,32:M=M+K:VPOKEM,ASC("o")
700 K=0
710 GOTO 490
720 X=A(0):FORR=0TOQ:VPOKEA(R),32
730 NEXT
740 RETURN
750 CLS
760 PRINT
770 PRINT"  YOU CAUGHT IT "
780 PRINT" IN A TIME OF ";INT(TIME/50);" SECONDS"
790 PRINT
800 PRINT
810 PRINT" PRESS ANY KEY TO RE-RUN."
820 A$=INKEY$
830 A$=INKEY$:IFA$=""THEN GOTO 830
840 GOTO 60
```

Monster Chase

Summary

The playing arena is similar to that of Mutant Attack, but this time, arm your reflexes and hone all your senses – the monsters are out to get you and there's no waiting round for you to make clever tactical decisions – outrun them, lead them into the mines or die. Kill all of these monsters, and their friends emerge, and believe me, they have a lot of friends ! These monsters are also fairly clever, as they think about each move for a while, hang back for the right tactical moment, and pounce.

Program Commentary

50-130   Sets up variables and screen.

360-470   Subroutine to define characters used in game.

480-650   Draws screen outline and mines.

660-720   Position the monsters. Print onto screen.

730-760   Positions man randomly.

770-980   Moves man if required, checks for hitting monsters, walls or mines.

990-1150 Does the same for the monsters, homing in on you.

1160-1220 Ends game if run into monster.

1190-1220 Resets screen if all monsters killed.

1230-1280 Resets man after running into wall.

1300-1340 Resets man after hitting mine.

1350-1480 Ends game, prints score, hi-score etc.

```
10 REM
20 REM MONSTER CHASE
30 REM
40 REM
50 CLS
60 RESTORE
70 PLAY"V9L32C5GAGAGEGCGAG"
80 SC=0
90 SCREEN C
100 GOSUB 360
110 COLOR 1,4,4
120 POKE&HF3DB,0
130 KEYOFF
140 PRINT
150 PRINT"      MONSTER CHASE"
160 PRINT
170 PRINT
180 PRINT
190 PRINT
200 PRINT"  YOU ARE THE MAN    a
210 PRINT"  YOUR AIM IS TO AVOID THE MONSTERS b"
220 PRINT"  WHILE AVOIDING THE MINES d"
230 PRINT"  AND THE EDGE OF THE SCREEN c"
240 PRINT
250 PRINT"  CONTACT WITH MINES AND THE EDGE LOSES YOU POINTS
260 PRINT"  BUT IT KILLS THE MONSTERS
270 PRINT
280 PRINT" USE Z X : / TO MOVE"
290 PRINT" LEFT,RIGHT,UP AND DOWN"
300 PRINT
310 PRINT
320 PRINT"  PRESS ANY KEY TO START"
330 A$=INKEY$:IFA$=""THEN GOTO 330
340 CLS
350 GOTO 480
360 REM  CHARACTERS
370 FORR=97 TO 100
380 FORT=0TO7
390 READS$:S=VAL("&B"+S$+"00")
400 VPOKE2048+(8*R)+T,S
410 NEXT
420 NEXT
430 DATA111111,100001,111111,001100,101101,101101,001100,010010
440 DATA001100,011110,010010,111111,101101,101101,001100,010010
450 DATA111111,111111,111111,111111,111111,111111,111111,111111
460 DATA000000,101101,011110,001100,001100,011110,101101,000000
470 RETURN
480 REM SET UP SCREEN
490 CLS
500 FORR=1TO700:NEXT
510 MD=0
520 FORR=3 TO 35
530 LOCATER,2:PRINT"c"
540 LOCATER,20:PRINT"c"
```

```
550 NEXT
560 FORR=3TO19
570 LOCATE3,R:PRINT"c"
580 LOCATE35,R:PRINT"c"
590 NEXT
600 FORR=1TO20
610 X=INT(RND(1)*32+3)
620 Y=INT(RND(1)*16+3)
630 IFVPEEK((Y*40)+X+1)<>32THENGOTO 610
640 LOCATEX,Y:PRINT"d"
650 NEXT
660 REM SET UP THE 4 MONSTERS
670 FORR=1TO4
680 X(R)=INT(RND(1)*32+3)
690 Y(R)=INT(RND(1)*16+3)
700 IFVPEEK((Y(R)*40)+X(R)+1)<>32THEN GOTO 680
710 LOCATEX(R),Y(R):PRINT"b"
720 NEXT
730 X=INT(RND(1)*32+3)
740 Y=INT(RND(1)*16+3)
750 IFVPEEK((Y*40)+X+1)<>32THENGOTO 730
760 LOCATE X,Y:PRINT"a":OX=X:OY=Y
770 REM MOVE MAN
780 TIME=0
790 UNT=INT(RND(1)*100)
800 A$=INKEY$
810 GOSUB 1490
820 IFA$=""THENA$=AA$
830 IFA$="Z"ORA$="z"THENX=X-1
840 IFA$="X"ORA$="x"THENX=X+1
850 IFA$="/"THENY=Y+1
860 IFA$=":"THENY=Y-1
870 AA$=A$
880 A=VPEEK((40*Y)+X+1)
890 IFA=98ORA=99ORA=100 THEN LOCATEOX,OY:PRINT" "
900 IFA=98 THEN GOTO 1160:LOCATEX,Y:PRINT"a":FORR=1TO1000:NEXT:REM HIT MONSTER
910 IFA=99 THEN GOSUB 1230:LOCATEX,Y:PRINT"a":FORR=1TO1000:NEXT:REM HIT WALL
920 IFA=100 THEN GOSUB 1300:LOCATEX,Y:PRINT"a":FORR=1TO1000:NEXT:REM HIT BLOCK"
930 LOCATEX,Y:PRINT"a"
940 IF(CX<>X)OR(OY<>Y)THEN LOCATEOX,OY:PRINT" ":OX=X:OY=Y
950 LOCATE3,0:PRINT"SCORE ";SC;"  HI-SCORE ";HS
960 IF MD=4 THEN GOTO 1190
970 MD=0
980 IF TIME<UNT THEN GOTO 800
990 REM MOVE MONSTERS
1000 MD=0
1010 FOR R=1 TO 4
1020 GOSUB 1490
1030 IF C(R)=1 THEN MD=MD+1:LOCATE X(R), Y(R):GOTO 1140
1040 LOCATE X(R),Y(R):PRINT" "
1050 IFX(R)<XTHENX(R)=X(R)+1
1060 IFX(R)>XTHENX(R)=X(R)-1
1070 IFY(R)<YTHENY(R)=Y(R)+1
1080 IFY(R)>YTHENY(R)=Y(R)-1
```

```
1090 VPE=VPEEK((Y(R)*40)+X(R)+1)
1100 IFVPE=100THEN SC=SC+5:C(R)=1:LOCATEX(R),Y(R):PRINT"d":GOTO 1140
1110 IFVPE=97THEN GOTO 1350:REM END OF LIFE
1120 LOCATE OX(R),OY(R):PRINT" "
1130 LOCATEX(R),Y(R):PRINT"b":OX=X:OY=Y
1140 NEXT
1150 GOTO 770
1160 REM HIT MONSTER
1170 PLAY"L32CBA"
1180 GOTO 1350:REM END OF LIFE
1190 PLAY"L32GAGAG"
1200 FORR=1TO4:C(R)=0:NEXT
1210 SC=SC+20
1220 GOTO 480
1230 SC=SC-1:PLAY"L32V9GG"
1240 X=INT(RND(1)*32+3)
1250 Y=INT(RND(1)*16+3)
1260 IFVPEEK((Y*40)+X+1)<>32THENGOTO 1240
1270 LOCATEX,Y:PRINT"d"
1280 RETURN
1290 GOTO 480
1300 SC=SC-10:PLAY"L32V9GG"
1310 X=INT(RND(1)*32+3)
1320 Y=INT(RND(1)*16+3)
1330 IFVPEEK((Y*40)+X+1)<>32THEN GOTO 1310
1340 RETURN
1350 CLS:PLAY"L64GABCDEF"
1360 PRINT
1370 PRINT" YOU ARE DEAD."
1380 PRINT
1390 PRINT"  YOU SCORED ";SC
1400 IFSC>HSTHENHS=SC:PRINT"  A NEW HIGH SCORE":GOTO1420
1410 PRINT "HIGH SCORE IS ";HS
1420 PRINT
1430 PRINT
1440 PRINT" PRESS ANY KEY TO PLAY AGAIN"
1450 FORR=1TO100:A$=INKEY$:NEXT
1460 A$=INKEY$
1470 A$=INKEY$:IFA$=""THEN GOTO 1470
1480 GOTO 50
1490 REM PLAY NOTE
1500 X$=CHR$(INT(RND(1)*7)+ASC("A"))
1510 PLAY"L64V5XX$;"
1520 RETURN
```

Journey

Summary

Fancy yourself as a hotshot driver ? Imagine driving down a thin canyon, trying desperately not to hit the walls and the boulders in your path. However, you are armed with a powerful laser beam which destroys all boulders in your path, at a cost of 10 points. You are also equipped with a 5th order battle defence screen, allowing you to crash 5 times before the inevitable end to the game.

Program Commentary

290-440   Set up characters for road, car and boulder.

500   Sets the screen height in lines. Any excess lines are lost from the bottom.

510-540   Controls movement of road.

560-590   Print the road. The score is then printed at the bottom, and the screen height set so that scrolling does not affect it.

600-630   Move car, fire if necessary.

640   Tests for crash.

650'   Randomly place boulder.

770-790   Clear boulders if laser fired.

```
10 REM
20 REM     JOURNEY
30 REM
40 CLS
50 KEYOFF
60 POKE&HF3DB,0
70 SCREEN 0
80 COLOR 1,4,7
90 PLAY"V5L32GABGABCGABCDEF"
100 PRINT
110 PRINT"        JOURNEY "
120 PRINT
130 PRINT
140 PRINT
150 PRINT"  THE AIM IS TO REMAIN ON THE"
160 PRINT"  ROAD AS LONG AS POSSIBLE"
170 PRINT
180 PRINT"  USE Z AND X TO STEER "
190 PRINT"  LEFT AND RIGHT."
200 PRINT
210 PRINT"  USE [RETURN] TO FIRE"
220 PRINT
230 PRINT"     GOOD LUCK"
240 PRINT
250 PRINT"  PRESS ANY KEY TO START."
260 A$=INKEY$:IFA$=""THENGOTO 260
270 SCREEN 0
280 SC=0:LI=5
290 RESTORE
300 FORR=0TO7
310 READS$:S=VAL("&B"+S$+"00")
320 VPOKE2048+(8*97)+R,S
330 NEXT
340 DATA111111,100001,111111,001100,101101,101101,001100,010010
350 FORR=0TO7
360 READS$:S=VAL("&B"+S$+"00")
370 VPOKE2048+(8*98)+R,S
380 NEXT
390 DATA000000,001100,011110,110011,110011,011110,001100,011110
400 FORR=0TO7
410 READS$:S=VAL("&B"+S$+"00")
420 VPOKE2048+(8*99)+R,S
430 NEXT
440 DATA111111,111111,111111,111111,111111,111111,111111,111111
450 CLS
460 ROAD$="c           c"
470 X=10:FORR=1TO22:PRINTTAB(X)ROAD$:NEXT
480 Q=15
490 REM SKI RUN BIT
500 POKE&HF3B1,22
510 RM=INT(RND(1)*2):IFRM=0THENRM=-1
520 X=X+RM
530 IFX<2THENX=2
540 IFX>18THENX=18
```

```
550 PLAY"L64V4M10S14A"
560 PRINT"ccccccccccccccccccccccccccccccccccc";
570 LOCATEX,21:PRINTROAD$
580 POKE&HF3B1,24:LOCATE0,23:PRINT"SCORE ";SC;"  LIVES ";LI+1;
590 POKE&HF3B1,22
600 A$=INKEY$
610 IFA$="Z"ORA$="z"THENQ=Q-1
620 IFA$="X"ORA$="x"THENQ=Q+1
630 IFA$=CHR$(13)THENGOSUB740
640 IFVPEEK((3*40)+Q+1)<>32THENGOSUB710
650 IFRND(1)>.5THENGOSUB720
660 ROAD$="c          c"
670 SC=SC+1
680 LOCATE Q,3:PRINT"a":LOCATEOQ,2:PRINT" ":OQ=Q
690 LOCATE1,21
700 GOTO 490
710 PLAY"V10L32GAGABBGA":IFLI=0THENGOTO810ELSELI=LI-1:RETURN
720 LOCATE X+1+INT(RND(1)*LEN(ROAD$)-1),20:PRINT"b"
730 RETURN
740 REM FIRE.
750 SC=SC-10:IFSC<0THENSC=0
760 PLAY"V7L32CDEF"
770 FORT=3TO10
780 IF VPEEK((T*40)+Q+1)<>ASC("c")THENLOCATEQ,T:PRINT" "
790 NEXT
800 RETURN
810 BEEP
820 CLS
830 PRINT
840 PRINT
850 PRINT"    YOU ARE DEAD."
860 PRINT
870 PRINT"    YOU HAVE SCORED ";SC;" POINTS."
880 PRINT
890 PRINT
900 PRINT"  DO YOU WANT ANOTHER GO (Y/N) ?"
910 A$=INKEY$:IFA$=""THENGOTO910
920 IFA$="Y"ORA$="y"THEN GOTO 40
930 IFA$="N"ORA$="n"THEN CLS:END
940 GOTO 910
```

Snake

Summary

A fast-action fun game for all the family. The idea of the game is to steer your snake through the grass, eating people as you go. For every person that you eat, a cross and a bottle of milk of magnesia appear. You must first drink the milk in order to prevent indigestion when you eat a cross. Unfortunately, your snake grows steadily longer as you eat things, and running into your own tail is fatal. The author accepts no responsibility for people starving to death due to addiction to the game - please eat and sleep occasionally !

Program Commentary

90-190   Set up characters.

200   Sets up screen arrays to store snake.

260-470   Draw screen, set up variables.

480-770   Controls the game itself. End if run into edge, own tail, or suffering from terminal indigestion.

780-990   Routines called from main program, for collisions with the various objects on screen.

1000-1140 Scoring and end of game.

```
10 REM
20 REM    SNAKE
30 REM
40 CLS
50 POKE&HF3DB,0:REM KEYCLICK OFF
60 COLOR 1,4,7
70 KEYOFF
80 SCREEN 0
90 FOR C=91 TO 94
100 FORR=0TO7:READS$:S=VAL("&B"+S$+"00"):VPOKE2048+(C*8)+R,S:NEXT
110 REM CHARACTER [\]^
120 NEXT
130 DATA111111,100001,111111,001100,101101,101101,001100,010010
140 DATA001100,011110,101101,111111,100001,111111,011110,001100
150 DATA111111,011110,011110,111111,111111,111111,111111,011110
160 DATA001100,001100,001100,110011,110011,001100,001100,001100
170 FORR=0TO7:VPOKE2048+(8*ASC("|"))+R,&B00110000:NEXT
180 FORR=0TO7:READS$:S=VAL("&B"+S$+"00"):VPOKE2048+(ASC("-")*8)+R,S:NEXT
190 DATA000000,000000,000000,111111,111111,000000,000000,000000
200 DIM A1(37,25),B1(37,21)
210 S=0
220 GOSUB 1150
230 CLS:LOCATE 0,10:PRINT"SPEED.. FAST,MEDIUM OR SLOW (1,2 OR 3)?"
240 A$=INKEY$:IFA$<"1"ORA$>"3"THEN GOTO 240
250 SP=VAL(A$)
260 CLS
270 NM=0
280 X=17:Y=12
290 A=X:B=Y
300 XM=0:YM=0
310 CLS
320 NM=NM+10
330 F=S
340 FORR=2TO36:LOCATER,0:PRINT"-":LOCATER,22:PRINT"-":NEXT
350 FORR=1 TO 21:LOCATE2,R:PRINT"|":LOCATE37,R:PRINT"|":NEXT
360 RESTORE
370 BEEP
380 FORZ=1 TO NM
390 Q=INT(RND(1)*33)+3:W=INT(RND(1)*20)+2
400 IF VPEEK((W*40)+Q+1)<>32 THEN GOTO 390
410 LOCATE Q,W:PRINT"["
420 NEXT
430 ST=0
440 Q=S
450 LOCATE2,0:PRINT"SCORE:"+STR$(S)
460 LOCATE23,0:PRINT"HI-SCORE:"+STR$(H)
470 LOCATE2,22:PRINT"STRENGTH:"+STR$(E)+"-"
480 A1(X,Y)=XM
490 B1(X,Y)=YM
500 X=X+XM
510 Y=Y+YM
520 PLAY"V5L50A"
530 AS=VPEEK((Y*40)+1+X)
540 IF X=2ORX=37ORY=0ORY=22 THEN GOTO 1000
```

```
550 IFAS=32 THEN GOTO 630
560 IFAS=92 THEN GOTO 1000
570 IFAS=94 AND ST=0 THEN GOTO 1000
580 Q=S
590 IF AS=91 THEN GOTO 780
600 IF AS=94 THENGOTO 880
610 IF AS=93 THEN GOTO 940
620 IFS-F=NM*70 THEN GOTO 280
630 LOCATEX,Y:PRINT"\"
640 Q=Q+10
650 IFQ<=STHEN GOTO 700
660 C=A1(A,B)
670 B=B+B1(A,B)
680 A=A+C
690 LOCATEA,B:PRINT" "
700 X$=INKEY$
710 IF SP=1 THEN GOTO 730
720 FOR D = 1 TO 10*(SP-1):NEXT
730 IFX$="X"ORX$="x" THEN XM=1:YM=0
740 IFX$="Z"ORX$="z" THENXM=-1:YM=0
750 IFX$="/"THENYM=1:XM=0
760 IFX$=":"THENYM=-1:XM=0
770 GOTO 480
780 S=S+10
790 PLAY"V9L32BCD"
800 LOCATE 8,0:PRINTS
810 FORG=1TO2
820 L=INT(RND(1)*34)+3
830 LL=INT(RND(1)*19)+3
840 IFVPEEK((40*LL)+1+L)<>32THENGOTO820
850 IFG=1THENLOCATEL,LL:PRINT"]"ELSELOCATEL,LL:PRINT"^"
860 NEXT
870 GOTO 620
880 S=S+40
890 PLAY"V9L32GAG"
900 LOCATE8,0:PRINTS
910 ST=ST-1
920 LOCATE10,22:PRINTST
930 GOTO 620
940 S=S+20
950 PLAY"V9L32DCB"
960 LOCATE8,0:PRINTS
970 ST=ST+1
980 LOCATE10,22:PRINTST
990 GOTO620
1000 PLAY"V9L32AGBDCG"
1010 C=B1(A,B)
1020 A=A+A1(A,B)
1030 B=B+C
1040 IF VPEEK((40*B)+A+1)<>92   THEN GOTO 1070
1050 LOCATE A,B:PRINT" "
1060 GOTO 1010
1070 LOCATE 16,11:PRINT"SCORE"
1080 IFS>HTHENLOCATE9,11:PRINT"NEW HI-":LOCATE26,11:PRINT"!!":H=S
```

```
1090 LOCATE 21,11:PRINTS
1100 LOCATE 14,14:PRINT"ANOTHER GO (Y/N) ?"
1110 A$=INKEY$:IFA$=""THENGOTO1110
1120 IFA$="Y"ORA$="y"THEN GOTO 210
1130 IFA$="N"ORA$="n"THEN CLS:PRINT:PRINT:END
1140 GOTO 1110
1150 CLS
1160 PRINT
1170 PRINT
1180 PRINT"     SNAKE ATTACK"
1190 PRINT
1200 PRINT
1210 PRINT"   YOU THE SNAKE ..... \"
1220 PRINT"ENJOY EATING PEOPLE... ["
1230 PRINT"AS A BONUS EVERY PERSON YOU"
1240 PRINT"EAT GIVES TWO THINGS;"
1250 PRINT" 1/    CROSSES"
1260 PRINT" 2/    INDIGESTION PILLS."
1270 PRINT" BEFORE EATING A CROSS IT IS"
1280 PRINT" ADVISABLE TO EAT ON OF THE LATTER."
1290 PRINT" THIS KEEPS YOUR STRENGTH UP."
1300 PRINT
1310 PRINT" USE Z  X  :  /  TO STEER"
1320 PRINT" LEFT,RIGHT,UP AND DOWN"
1330 PRINT
1340 PRINT "     GOOD LUCK."
1350 PRINT
1360 PRINT" PRESS ANY KEY TO START"
1370 A$=INKEY$:IFA$=""THEN GOTO 1370
1380 RETURN
```

Bike Chase

Summary

You are in charge of the latest high-speed rocket bike which you must steer around the screen without bumpping into into the vapour trails left behind, or the edge of the playing arena. Also there are a few obstacles scattered about to add to the fun.

You can play an opponent, or practise solo as you dart and weave your way around (or into) the hazards on the screen. The longer you survive, the higher your score. The game ends when either player crashes.

If, before playing, you choose not to see the instructions then your previous choices of number of players, difficulty, etc., are saved in function key 1. When you play again, press F1 to repeat the same options.

Program Commentary

120-150	Ask for instructions, print if required.
160-190	Level of difficulty controls number of obstacles.
200-250	Input number of players - 1 or 2.
290-350	Re-seed RNG, set up hi-res screen for text, play starting tune, clear keyboard buffer.
360-560	Main program loop -
370-380	Set direction strings, get player's input.
390-470	Change 1 or 2 direction strings as appropriate.
480-520	Use strings from above to move bikes.
530-540	Test for any collisions for both players.
550-560	Plot bike positions, return to start of loop if no crashes.
570-610	Set text screen, play 'dead' tune, print score.
620-640	Clear keyboard buffer, wait for space bar to repeat game.
650-670	Screen 0 centered text routine. String in P$, vertical position in YP%.
680-870	Text for instructions.

```
10 REM *******************
20 REM **              **
30 REM ** Missile Chase **
40 REM **              **
50 REM *******************
60 REM
70 REM
80 K$=" "
90 SCREEN0:KEYOFF:TROFF
100 COLOR15,0,0
110 FORA%=0TO40:I$=INKEY$:NEXT
120 P$="Do you want instructions ? (Y/N)":YP%=10:GOSUB660
130 I$=INKEY$:IFI$=""ORINSTR("YNyn",I$)=0THEN130
140 IFI$="Y"ORI$="y"THENCLS:U%=-1:GOSUB690ELSEU%=0
150 K$=K$+I$
160 CLS:INPUT"Level of difficulty ( 1 - 50 )";L%
170 IFL%<1ORL%>50THENBEEP:PRINT:PRINT"Outside range.Press SPACE BAR and try  again"
180 IFL%<1ORL%>50THENI$=INKEY$:IFI$<>" "THEN180ELSE160
190 K$=K$+STR$(L%)+CHR$(13)
200 X1%=50:X2%=200:Y1%=90:Y2%=90
210 CLS
220 P$="1 or 2 player game (1/2) ? ":YP%=1:GOSUB660
230 I$=INKEY$:IFI$<>"1"ANDI$<>"2"THEN230
240 IFI$="1"THENN%=0ELSEN%=-1
250 K$=K$+I$
260 IFU%THENKEY1,""ELSEKEY1,K$
270 K$=" "
280 P%=0
290 T=RND(-TIME)
300 SCREEN2
310 OPEN"GRP:"AS#1
320 FORA%=1TOL%:PRESET(RND(1)*250,RND(1)*190):PRINT#1,"*":NEXT
330 PLAY"V15O3L64CEGCGEC"
340 LINE(0,0)-(255,0):DRAW"M255,191M0,191M0,0"
350 TIME=0
360 REM START OF MAIN LOOP
370 S1$="RU":S2$="LD"
380 I$=INKEY$
390 IFI$="A"THENS1$=LEFT$(S1$,1)+"U"
400 IFI$="Z"THENS1$=LEFT$(S1$,1)+"D"
410 IFI$="X"THENS1$="L"+RIGHT$(S1$,1)
420 IFI$="C"THENS1$="R"+RIGHT$(S1$,1)
430 IFNOTN%THEN480
440 IFI$=":"THENS2$=LEFT$(S2$,1)+"U"
450 IFI$="/"THENS2$=LEFT$(S2$,1)+"D"
460 IFI$=","THENS2$="L"+RIGHT$(S2$,1)
470 IFI$="."THENS2$="R"+RIGHT$(S2$,1)
480 IFLEFT$(S1$,1)="R"THENX1%=X1%+1ELSEX1%=X1%-1
490 IFRIGHT$(S1$,1)="D"THENY1%=Y1%+1ELSEY1%=Y1%-1
500 IFNOTN%THEN530
510 IFLEFT$(S2$,1)="R"THENX2%=X2%+1ELSEX2%=X2%-1
520 IFRIGHT$(S2$,1)="D"THENY2%=Y2%+1ELSEY2%=Y2%-1
530 IFPOINT(X1%,Y1%)<>0THENP%=1
540 IFN%ANDPOINT(X2%,Y2%)<>0THENP%=2
```

```
550 PSET(X1%,Y1%):IFN%THENPSET(X2%,Y2%)
560 IFP%=0THEN380
570 SCREEN0
580 PLAY"V1503L16CO2BAGFEDC"
590 P$="You´re dead player "+STR$(P%):YP%=5:GOSUB660
600 P$="Your brave attempt lasted "+STR$(INT(TIME/50))+" seconds.":YP%=10:GOSUB660
610 P$="Press SPACE BAR for another game":YP%=15:GOSUB660
620 FORA%=0TO32:I$=INKEY$:NEXT
630 I$=INKEY$:IFI$<>" "THEN630
640 CLEAR:GOTO90
650 REM CENTRED TEXT ROUTINE
660 LOCATE19-LEN(P$)/2,YP%:PRINTP$;
670 RETURN
680 REM INSTRUCTION TEXT
690 P$="The object of this game is to steer":YP%=0:GOSUB660
700 P$="your high power rocket around the":YP%=1:GOSUB660
710 P$="screen while avoiding obstacles on":YP%=2:GOSUB660
720 P$="the screen,and the vapour trails of":YP%=3:GOSUB660
730 P$="toxic exhaust fumes of your and,if":YP%=4:GOSUB660
740 P$="selected,your opponent´s rockets.":YP%=5:GOSUB660
750 P$="Use the following keys :":YP%=7:GOSUB660
760 P$="   Up - PLAYER1 A,PLAYER2 :":YP%=9:GOSUB660
770 P$=" Down - PLAYER1 Z,PLAYER2 /":YP%=10:GOSUB660
780 P$=" Left - PLAYER1 X,PLAYER2 ,":YP%=11:GOSUB660
790 P$="Right - PLAYER1 C,PLAYER2 .":YP%=12:GOSUB660
800 P$="You must try to avoid the edges of the":YP%=14:GOSUB660
810 P$="screen and each others´ trails,":YP%=15:GOSUB660
820 P$="although you might be lucky and hit":YP%=16:GOSUB660
830 P$="a vapour pocket too thin to damage you":YP%=17:GOSUB660
840 P$="and thus survive,though this":YP%=18:GOSUB660
850 P$="technique tends to become unreliable":YP%=19:GOSUB660
860 P$="just at the wrong time !!!":YP%=20:GOSUB660
870 P$="Press SPACE BAR to continue":YP%=23:GOSUB660
880 FORA%=0TO32:I$=INKEY$:NEXT
890 FORA%=0TO32:I$=INKEY$:NEXT
900 I$=INKEY$:IFI$<>" "THEN900
910 RETURN
```

Dice

Summary

Small but sweet the expression goes. This ´shorty´ rolls any number of six-sided dice (the usual sort), and then analyses the results by printing the number of ones, twos etc. rolled. It demonstrates how to use the random number generator to produce integers between given limits, and a simple counting method.

Program Commentary

30-50	Set up the screen for 40 characters, black text on blue background.
60	Sets a random seed for the dice.
70-130	Display the title and program aim.
140-150	Find out the number of dice to be ´rolled´.
170-370	Select and display the ´rolls´ one by one, showing the total of each number rolled.
380-430	Restart the program.

```
10 REM
20 REM DICE ROLLER
30 CLS
40 SCREEN0
50 COLOR 1,4,7
60 SEED=RND(-TIME)
70 PRINT:PRINT" DICE ROLLER"
80 PRINT " -----------"
90 PRINT:PRINT
100 PRINT" THIS PROGRAM ROLLS A NUMBER"
110 PRINT"OF 6 SIDED DICE AND PRINTS"
120 PRINT"THE RESULTS."
130 PRINT
140 INPUT" HOW MANY DICE DO YOU WANT THROWN";N
150 IFN<1THENGOTO140
160 PRINT:PRINT:PRINT" PRESS ANY KEY TO ROLL"
170 A$=INKEY$
180 A$=INKEY$:IFA$=""THENGOTO180
190 CLS
200 PRINT:PRINT
210 PRINT"ONES...."
220 PRINT"TWOS...."
230 PRINT"THREES.."
240 PRINT"FOURS..."
250 PRINT"FIVES..."
260 PRINT"SIXES..."
270 PRINT:PRINT"TOTAL..."
280 PRINT:PRINT:PRINT "    D I C E   R O L L E R"
290 FORR=1TO6:A(R)=0:NEXT
300 FORR=1TON
310 Q=INT(RND(1)*6)+1
320 A(Q)=A(Q)+1
330 IF A(Q)>HTHENLOCATE9,A+1:PRINT" ":A=Q:H=A(Q)
340 LOCATE9,A+1:PRINT">"
350 LOCATE10,Q+1:PRINTA(Q)
360 LOCATE10,9:PRINTR
370 NEXT
380 LOCATE5,20
390 PRINT" PRESS ANY KEY TO RUN"
400 FORR=1TO50:A$=INKEY$:NEXT
410 A$=INKEY$
420 A$=INKEY$:IFA$=""THENGOTO420
430 RUN
```

Card Deal

Summary

Just as you would expect this program is more of a
subroutine than a whole program, designed to be included in a
more complex creation such as a blackjack swindling program or
whatever else your fiendish imagination can cook up for it. It
takes a pack of cards in its memory, whose number and composition
can be easily altered and rearranges them randomly. One wonders
what the effect would be on a Tarot pack when shuffled by
computer - whose future would it read...?

Program Commentary

40-80   Set up the screen for 40 characters, black text on
           green background (like a card table).

90-170   Display the title and objective of the program.

180-190   Set up the program for replacing or discarding cards.

200-280   Set up the pack of cards for play.

290-320   See how many cards are required. This includes
           validation routines to stop such cheats as discarding,
           and still drawing 53 cards from the pack.

330-420   Choose the cards randomly. If cards are discarded,
           lines 370 and 390 check for this and discard used cards
           as necessary. (Try saying that after you've had too
           many!)

```
10 REM
20 REM CARD DEAL
30 REM
40 CLS
50 KEYOFF
60 DIMA$(4,13),B$(4)
70 SCREEN0
80 COLOR 1,2,7
90 PRINT:PRINT:PRINT"   CARD DEALER"
100 PRINT      "   -----------"
110 PRINT:PRINT:PRINT
120 PRINT"   THE PROGRAM WILL SELECT AT RANDOM"
130 PRINT"   A CARD "
140 PRINT:PRINT" YOU MAY SELECT WHETHER CARDS"
150 PRINT"ARE REPLACED AFTER SELECTION OR"
160 PRINT"DISCARDED."
170 PRINT
180 INPUT" REPLACE OR DISCARD (R/D)";Q$
190 IFQ$="R"ORQ$="D"THENGOTO200 ELSE 180
200 CLS
210 REM SET UP PACK
220 DATA CLUBS,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,KNAVE,QUEEN,KING
230 DATA HEARTS,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,KNAVE,QUEEN,KING
240 DATA SPADES,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,KNAVE,QUEEN,KING
250 DATA DIAMONDS,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE,TEN,KNAVE,QUEEN,KING
260 FORT=1TO4:READB$(T):FORR=1TO13
270 READA$(T,R)
280 NEXTR,T
290 PRINT:PRINT:PRINT:PRINT" HOW MANY CARDS (1 TO 52 IF DISCARDING";
300 INPUT"   ,1 OR MORE IF REPLACING.)";NC
310 IFNC<1THENGOTO290
320 IFNC>52ANDQ$="D"THENGOTO290
330 FOR CARDS=1 TO NC
340 REM SELEB3 A CARD
350 Q=RND(-TIME)
360 T=INT(RND(1)*4)+1:R=INT(RND(1)*13)+1
370 IFA$(T,R)="   "THENGOTO350
380 IFA$(T,R)="   "THENGOTO350
390 PRINTCARDS;"      ";A$(T,R);" OF ";B$(T)
400 IFQ$="D"THENA$(T,R)="   "
410 BEEP
420 NEXT
430 PRINT
440 PRINT
450 PRINT"    THAT´S YOUR LOT."
460 PRINT
470 PRINT"   PRESS ANY KEY TO CONTINUE."
480 PRINT
490 FORR=1TO100
500 A$=INKEY$
510 NEXT
520 A$=INKEY$
530 IFA$=""THENGOTO520
540 CLS:BEEP:RUN
```

Word Sort

Summary

This program uses a version of the Shell-Metzner sort method. This is a fast routine, offering considerable speed advantages over bubble sorts and binary chop sorts especially when a large number of items are involved.

The world of sorting is a complex mathematical place inhabited by computer scientists and other fearsome creatures, who can spend years knocking a couple of nanoseconds off processing time. However, the average user is quite happy to use the result and ignore the thought behind it, which is the point of this program. No serious programmer would ever consider writing a sort, shuffle, or search routine as the best ones have all been written. There is no point re-inventing the wheel, especially not if yours is going to be square!

Program Commentary

40-60   Screen set-up, 40 characters, white text, blue back.

70-150   Display program title and aim.

160-250   Input information. As the clear command (used later to reserve just enough string space) also destroys all variables, it is necessary to store the one variable so far set, the number of words. This is stored in free memory at &hF3AF.

260-390   These give the options for the main part of the program - sort or edit the word list, or restart.

400-550   Sort the data, using a modified Shell-Metzner sort routine. (It sounds mathematical, and it is!)

560-740   Display the data either to screen or printer, using LPRINT.

750-940   Display, and if necessary, alter the information. This routine is a modification of the input routine in lines 220-250.

```
10 REM
20 REM   WORD SORT
30 REM
40 SCREEN 0
50 KEYOFF
60 COLOR 15,4,7
70 PRINT:PRINT:PRINT"       ALPHABETICAL SORT"
80 PRINT"       ------------------"
90 PRINT
100 PRINT
110 PRINT"  THIS PROGRAM SORTS SEVERAL WORDS OR   PHRASES";
120 PRINT" INTO ALPHABETICAL ORDER USING    THE SHELL-METZNER METHOD"
130 PRINT"THIS METHOD INCREASES IN EFFICIENCY     WITH THE NUMBER OF ITEMS USED"
140 PRINT
150 PRINT
160 INPUT" HOW MANY ITEMS ";N
170 IFN<1THEN GOTO 160
180 POKE&HF3AF,N
190 CLEAR (N*100)
200 DIM A$(PEEK(&HF3AF))
210 N=PEEK(&HF3AF)
220 FORR=1 TO N
230 CLS:PRINT:PRINT:PRINT:PRINT
240 PRINTR;" ";:INPUTA$(R)
250 NEXT
260 CLS
270 PRINT:PRINT
280 PRINT"  1  SORT"
290 PRINT"  2  CORRECT"
300 PRINT"  3  RESTART"
310 PRINT
320 PRINT"      WHICH OPTION ?"
330 FORR=1TO20:S$=INKEY$:NEXT
340 S$=INKEY$:IFS$=""THENGOTO340
350 IFS$="1" THEN CLS:GOTO 400
360 IFS$="2" THEN GOTO 750     .
370 IFS$="3" THEN RUN
380 BEEP
390 GOTO 340
400 PRINT:PRINT:PRINT" PLEASE WAIT...SORTING"
410 M=N
420 M=INT(M/2)
430 IFM=0 THEN GOTO 550
440 K=N-M
450 J=1
460 I=J
470 L=I+M
480 IFA$(I)<A$(L)THEN GOTO 520
490 T$=A$(I):A$(I)=A$(L):A$(L)=T$
500 I=I-M
510 IFI<1 THEN GOTO 520 ELSE 470
520 J=J+1
530 IFJ>K THEN GOTO 420
540 GOTO 460
```

```
550 REM ALL SORTED
560 CLS
570 PRINT:PRINT:PRINT
580 PRINT"  1   OUTPUT TO SCREEN"
590 PRINT"  2   OUTPUT TO PRINTER"
600 FORR=1 TO 20:S$=INKEY$:NEXT
610 S$=INKEY$:IFS$="" THEN GOTO 610
620 IFS$="1" THEN GOTO 650
630 IFS$="2" THEN GOTO 710
640 BEEP:GOTO 620
650 FORR=1 TO N
660 PRINTR;" ";A$(R)
670 IFR/20=INT(R/20)THEN FORD=1 TO 2000:NEXT:BEEP
680 NEXT
690 FORD=1 TO 2000:NEXT:BEEP
700 GOTO 260
710 FORR=1 TO N
720 LPRINTR;" ";A$(R)
730 NEXT
740 GOTO 260
750 CLS
760 PRINT:PRINT
770 R=1
780 IFR>N THEN R=1
790 IFR<1 THEN R=N
800 CLS
810 PRINT:PRINT:PRINT:PRINTR;" ";A$(R)
820 PRINT:PRINT" 1 TO CORRECT"
830 PRINT" 2 TO ADVANCE"
840 PRINT" 3 TO GO BACK"
850 PRINT" 4 TO EXIT"
860 FORT=1 TO 10:S$=INKEY$:NEXT
870 S$=INKEY$:IFS$=""THEN GOTO 870
880 S=VAL(S$)
890 IFS<1 OR S>4 THEN BEEP:GOTO 870
900 ON S GOTO 910,920,930,940
910 PRINT:PRINT:PRINTR;" ";:INPUTA$(R)
920 R=R+1:GOTO 780
930 R=R-1:GOTO 780
940 GOTO 260
```

Mark Sort

Summary

A useful program for the up-to-date but harrassed teachers out there, this will sort the marks for a group of students and then display the results on the screen or printer. Being written by a statistician though it does not just stop at that. Initially it displays the upper quartile (top 25% of marks), but the program is easily altered as below to display other quartiles. Also it produces a mean (average) and standard deviation of the results as a basic analysis.

Naturally, the computer does not know what the data you are entering is, and so it could equally be used for telephone numbers, restaurant bills (a form of bistromathics ?) or anything else. The results are unlikely to be meaningful - the average of your friends' telephone numbers is not likely to give you directory enquiries - but it's all good fun anyway.

Program Commentary

40-60   Set up screen for the display.

100   Requests the number of marks to be input.

110-170   Input the values of the marks.

180-460   Sort the marks, and extract values for mean (average to the layman) and standard deviation. These are then rounded to two places of decimals by using:

$$X=INT(X*100+0.5)/100$$

420-460   Count the number of marks below the 40% mark.

470-640   These then display the information on the screen.

650-770   Routine to change incorrect values.

780-950   Display information to printer.

To change the displayed quartile, simply change the limits of the 'FOR' loop in line 590. e.g for the lower median quartile, change the limits to N/2 TO 3*N/4.

```
10 REM
20 REM MARK SORT
30 REM
40 SCREEN 0
50 COLOR 15,4,4
60 KEYOFF
70 PRINT:PRINT
80 PRINT"    MARK PROCESSOR"
90 PRINT:PRINT:PRINT
100 INPUT" HOW MANY MARKS IN THE SET";N
110 DIM I(N+5),D(N+5)
120 PRINT:PRINT:PRINT
130 PRINT" ENTER EACH MARK FOLLOWED BY RETURN"
140 PRINT
150 FORR=1 TO N
160 PRINTR;">";:INPUT I(R)
170 NEXT
180 FOR R =1 TO N STEP 10
190 X=R:Y=R+9
200 GOSUB 650
210 IF N<Y+1 THEN GOTO 230
220 NEXT
230 FOR P=1 TON
240 T=T+I(P)
250 NEXT
260 M=T/N
270 FORP=1 TO N
280 D(P)=I(P)-M
290 S=S+D(P)*D(P)
300 NEXT
310 SD=SQR(S/N)
320 M=INT(M*100+.5)/100
330 SD=INT(SD*100+.5)/100
340 PRINT" BUBBLE SORT IN ACTION"
350 F=0
360 FOR J=1 TO N
370 IF I(J)>=I(J+1) THEN 400
380 Z=I(J):I(J)=I(J+1):I(J+1)=Z
390 F=1
400 NEXT
410 IF F>0 THEN GOTO 350
420 FOR G=1 TO N
430 IF I(G)<I(1)*.4 THEN GOTO 450
440 GOTO 460
450 Q=Q+1
460 NEXT
470 GOSUB 500
480 INPUT" DO YOU WANT A PRINT OUT (YES OR NO)";A$
490 IFA$="YES" THEN GOTO 780 ELSE GOTO 920
500 CLS
510 FOR J=1 TO N:PRINTI(J);;
520 NEXT:PRINT:PRINT
530 PRINT" MEAN VALUE  =";M
540 PRINT
```

```
550 PRINT" STANDARD DEVIATION =";SD
560 PRINT
570 PRINT" UPPER QUARTILE "
580 PRINT
590 FOR W=1 TO N/4
600 PRINTI(W);;
610 NEXT
620 PRINT:PRINT
630 PRINT"NUMBER WITH MARKS BELOW 40%";Q:PRINT
640 RETURN
650 CLS
660 FOR P=X TO Y
670 PRINTP,I(P)
680 IF P=N THEN GOTO 710
690 NEXT
700 PRINT
710 INPUT" ARE THESE VALUES CORRECT (YES OR NO)";A$
720 IFA$="NO" THEN GOTO 740
730 GOTO 770
740 PRINT" TYPE THE CORRECT VALUE AS No, Value [RETURN]"
750 INPUT P,I(P)
760 GOTO 650
770 RETURN
780 LPRINT CHR$(12)
790 FOR J=1 TO N:LPRINTI(J);;
800 NEXTJ:LPRINT:LPRINT
810 LPRINT" MEAN VALUE  =";M
820 LPRINT
830 LPRINT" STANDARD DEVIATION =";SD
840 LPRINT
850 LPRINT" UPPER QUARTILE "
860 LPRINT
870  FOR W=1 TO N/4
880 LPRINTI(W);;
890 NEXT
900 LPRINT:LPRINT
910 LPRINT"NUMBER WITH MARKS BELOW 40%";Q:LPRINT
920 PRINT:PRINT:PRINT" PRESS ANY KEY TO RUN AGAIN"
930 FORR=1 TO 50:A$=INKEY$:NEXT
940 A$=INKEY$:IFA$="" THEN GOTO 940
950 RUN
```

Personality

Summary

Are you sitting comfortably ? Perhaps you would like to lie down on this couch ? Dr. Freud may not have approved, but this program sets out to test your personality by two methods - numerology and subconcious relation. Using these techniques, and information supplied by you it attempts to analyse you. It can be taken seriously, as the program is based on genuine psychoanalytical techniques, in which case all answers must be scrupulously honest for any bearing to be given to the results. Alternatively, it can be used in a more light-hearted vein, and tried on friends or relatives. You never know, it may reveal some things you weren't previously aware of, whether good or otherwise!

Program Commentary

10-30   Set up Screen 0 and colours for text,

40-120   Print out title and program aim.

130-180   Select type of test as explained above.

210-300   Print out the numerology aims and instructions.

340-480   Perform the analysis on the string input.

490-520   Call the various printout routines in the following lines (530,600,670,740,790,840,910,980,1030,1090,1160)

1230-1360 This is the second type of analysis. Here the question and answer type aim and instructions are printed. The individual questions are at lines 1370,1410,1450, 1490,1550,1610,1650,1710,1750.

1780-1870 Print out the analysis.

1880-1940 A routine called by both sections to set the user thinking...

```
10 CLS:COLOR 1,4,7
20 KEYOFF
30 SCREEN 0
40 PRINT:PRINT:PRINT"   PERSONALITY TESTER."
50      PRINT"   ------------------"
60 PRINT:PRINT
70 PRINT" THIS IS A PROGRAM DESIGNED TO"
80 PRINT" TEST YOUR PERSONALITY IN ONE OF"
90 PRINT" TWO WAYS..."
100 PRINT" 1/  NUMEROLOGY"
110 PRINT" 2/  QUESTION AND ANSWER "
120 PRINT:PRINT
130 PRINT" PRESS THE NUMBER OF YOUR CHOICE"
140 A$=INKEY$
150 A$=INKEY$:IFA$="1"ORA$="2"THENGOTO160 ELSE 150
160 REM THIS SELECTS THE OPTION
170 V=VAL(A$)
180 ON V GOTO 200,1230
190 RUN
200 REM  TEST BY NUMEROLOGY
210 CLS
220 PRINT:PRINT:PRINT
230 PRINT" THIS PROGRAM ANALYSES YOUR PERSONALITY"
240 PRINT" BY USING THE SCIENCE OF MATHEMATICS."
250 PRINT
260 PRINT" BY SIMPLY ENTERING YOUR NAME THE"
270 PRINT" COMPUTER WILL ´CALCULATE´ YOUR"
280 PRINT"      PERSONALITY"
290 PRINT
300 PRINT
310 DIMB(26)
320 INPUT" WHAT IS YOUR NAME ";N$
330 IFLEN(N$)<1THENPRINT"TRY AGAIN PLEASE.":GOTO320
340 DATA1,2,3,4,5,6,7,8,9,1,2,3,4,5,6,7,8,9,1,2,3,4,5,6,7,8
350 RESTORE
360 FORR=1TO26:READB(R):NEXT
370 T=0
380 FORR=1 TO LEN(N$)
390 A=ASC(MID$(N$,R,1))-64:IFMID$(N$,R,1)=" "THENGOTO420
400 IFA<0ORA>27THENBEEP:PRINT"ERROR CAPITALS ONLY":GOTO320
410 T=T+B(A)
420 NEXT
430 IF T=11ORT=22ORT<=9THENGOTO490
440 T$=STR$(T)
450 Q=0
460 FORR=1TOLEN(T$)
470 Q=Q+VAL(MID$(T$,R,1))
480 NEXT:T=Q:GOTO430
490 ON T GOTO 530,600,670,740,790,840,910,980,1030
500 IFT=11THENGOTO1090
510 IFT=22THENGOTO1160
520 PRINT"ERROR ":BEEP:GOTO320
530 CLS
540 PRINT"No ONE.A STRONG NUMBER,REPRESENTING"
```

```
550 PRINT"FORCE,CREATIVE ABILITY,INDIVIDUALISM,"
560 PRINT"AND SELF ASSERTIVENESS.THE BEARER OF"
570 PRINT"THIS NUMBER MUST BE CAREFUL NOT TO"
580 PRINT"BECOME TOO OFFICIOUS AND DOMINATING."
590 GOTO 1870
600 CLS
610 PRINT"No TWO.A SOCIAL NUMBER.TACT,DIPLOMACY,"
620 PRINT"AND THE ABILITY TO MAKE FRIENDS WILL"
630 PRINT"ENSURE ITS OWNER HAPPINESS.IT IS A"
640 PRINT"NUMBER OF THE HEART RATHER THEN THE"
650 PRINT"MIND."
660 GOTO 1870
670 CLS
680 PRINT"No THREE.MIRTH,GOOD CHEER,PLEASURE,"
690 PRINT"AND INSPIRATION ARE THE PRINCIPLE"
700 PRINT"CHARACTERISTICS OF THIS NUMBER.IT IS"
710 PRINT"THE NUMBER OF ORATORS,ACTORS,SINGERS,"
720 PRINT"DANCERS AND PAINTERS."
730 GOTO 1870
740 CLS
750 PRINT"No FOUR.THE NUMBER OF THE HARD WORKER,"
760 PRINT"THE PLODDER,BRINGER OF POVERTY,DRUDGERY";
770 PRINT"AND LACK OF INITIATIVE AND          UNDERSTANDING."
780 GOTO1870
790 CLS
800 PRINT"No FIVE.PEOPLE RULED BY FIVE ACCOMPLISHLITTLE BUT LEAD A CONFUSED,UNSTABLE"
810 PRINT"EXISTANCE.FIVE IS THE SYMBOL OF THE"
820 PRINT"ROLLING STONE."
830 GOTO1870
840 CLS
850 PRINT"No SIX.THOSE OWNING THIS NUMBER ARE THE";
860 PRINT"BACKBONE OF THE NATION.THEY ARE STAUNCH";
870 PRINT"FRIENDS,GOOD CITIZENS,HONEST RELIABLE"
880 PRINT"PEOPLE,AND HOME MAKERS.IT IS THE"
890 PRINT"MARRIAGE NUMBER."
900 GOTO 1870
910 CLS
920 PRINT"No SEVEN.THOSE RULED BY THIS NUMBER"
930 PRINT"HAVE A HARD TIME IN THIS WORLD AND ARE"
940 PRINT"NOT DESTINED FOR RENOWN OR GLORY."
950 PRINT"THEIR NATURES ARE FULL OF BEAUTY AND "
960 PRINT"POETRY,AND TEND TOWARDS THE OCCULT."
970 GOTO 1870
980 CLS
990 PRINT"No EIGHT.THE NUMBER OF BUSINESS SUCCESS,ACHIEVEMENT,ATTAINMENT";
1000 PRINT",AND ACQUIREMENT.MONEY AND SUCCESS ARE THE QUALITIES"
1010 PRINT"THIS NUMBER ATTRACTS."
1020 GOTO 1870
1030 CLS
1040 PRINT"No NINE.THIS IS A NUMBER OF GREAT"
1050 PRINT"SUCCESS FINANCIALLY BUT NOT OF PRACTIC-ALITY.GENIUS AND GREAT TALENT ARE"
1060 PRINT"THE QUALITIES OF THIS NUMBER.ITS "
1070 PRINT"INFLUENCE IS UNIVERSAL."
1080 GOTO 1870
```

```
1090 CLS
1100 PRINT"No ELEVEN.THIS IS THE NUMBER OF GENIUS.";
1110 PRINT"THESE PEOPLE SEEN TO BE ENLIGHTENED BY AN INNER FIRE.";
1120 PRINT"THEY ARE MARKED BY"
1130 PRINT"SERENITY UNDER ALL CIRCUMSTANCES,NO"
1140 PRINT"MATTER HOW ADVERSE."
1150 GOTO 1870
1160 CLS
1170 PRINT"No TWENTY-TWO.THIS NUMBER HAS BROUGHT"
1180 PRINT"TO MANKIND GREAT REFORMERS.GREAT"
1190 PRINT"LEADERS FOR GOOD.IF THIS NUMBER IS     GIVEN ITS DESTRUCTIVE EXPRESSION,";
1200 PRINT"       HOWEVER,IT RUNS INTO BIGOTRY,RELIGIOUS MANIA,AND EVEN INTO ";
1210 PRINT"MISERLINESS"
1220 GOTO1870
1230 REM   TEST BY QUESTION AND ANSWER
1240 CLS
1250 PRINT:PRINT:PRINT
1260 PRINT" THIS PROGRAM ANALYSES YOUR PERSONALITY"
1270 PRINT" BY USING A SERIES OF MULTIPLES CHOICE
1280 PRINT" QUESTIONS."
1290 PRINT" PRESS ANY KEY TO START"
1300 A$=INKEY$
1310 A$=INKEY$:IFA$=""THENGOTO1310
1320 CLS
1330 PRINT
1340 PRINT"  YOU MUST ANSWER WITHOUT THOUGHT."
1350 PRINT" LET YOUR SUBCONCIOUSNESS GUIDE YOU."
1360 PRINT
1370 INPUT " ON ENTERING A FOREST IS IT LIGHT OR    DARK (1 OR 2)";Q$
1380 IFQ$="1"ORQ$="2"THENGOTO1390ELSE GOTO1370
1390 IFQ$="1"THENA$(1)="PLEASING"ELSEA$(1)="DEPRESSING"
1400 CLS:PRINT
1410 INPUT" IS THE PATH WIDE OR NARROW (1 OR 2)";Q$
1420 IFQ$="1"ORQ$="2"THENGOTO1430ELSE GOTO1410
1430 IFQ$="1"THENA$(2)="EASING"ELSEA$(2)="FORCING"
1440 CLS:PRINT
1450 INPUT" IS THE PATH TWISTING OR STRAIGHT (1 OR 2)";Q$
1460 IFQ$="1"ORQ$="2"THENGOTO1470ELSE GOTO1450
1470 IFQ$="1"THENA$(3)="UNCERTAIN"ELSEA$(3)="CERTAIN"
1480 CLS:PRINT
1490 INPUT" YOU ENCOUNTER AN ANIMAL IS IT LARGE,   MEDIUM OR SMALL (1,2 OR 3)";Q$
1500 IFQ$="1"ORQ$="2"ORQ$="3"THENGOTO1510ELSE GOTO1490
1510 IFQ$="1"THENA$(4)="UP"
1520 IFQ$="2"THENA$(4)="EQUALLY"
1530 IFQ$="3"THENA$(4)="DOWN"
1540 CLS:PRINT
1550 INPUT" WHAT IS YOUR REACTION - HIDE,ATTACK OR MAKE FRIENDS (1,2 OR 3)";Q$
1560 IFQ$="1"ORQ$="2"ORQ$="3"THENGOTO1570ELSE GOTO1550
1570 IFQ$="1"THENA$(5)="SHY"
1580 IFQ$="2"THENA$(5)="AGGRESSIVE"
1590 IFQ$="3"THENA$(5)="FRIENDLY"
1600 CLS:PRINT
1610 INPUT" YOU COME TO A BODY OF WATER.IS IT A    POND OR A STREAM (1 OR 2)";Q$
1620 IFQ$="1"ORQ$="2"THENGOTO1630ELSE GOTO1610
```

```
1630 IFQ$="1"THENA$(6)="PLACID"ELSEA$(6)="LIVELY"
1640 CLS:PRINT
1650 INPUT"IS THE WATER DEEP AND DARK,CLEAR AND     SPARKLING OR STAGNANT (1,2 OR 3)";Q$
1660 IFQ$="1"ORQ$="2"ORQ$="3"THENGOTO1670ELSE GOTO1650
1670 IFQ$="1"THENA$(7)="DEEP"
1680 IFQ$="2"THENA$(7)="CLEAN"
1690 IFQ$="3"THENA$(7)="EVIL"
1700 CLS:PRINT
1710 INPUT" YOU HAVE TO PASS THE WATER.DO YOU WALK AROUND OR ACROSS IT (1 OR 2)";Q$
1720 IFQ$="1"ORQ$="2"THENGOTO1730ELSE1710
1730 IFQ$="1"THENA$(8)="INACTIVE"ELSEA$(8)="ACTIVE"
1740 CLS:PRINT
1750 INPUT" YOU FIND A KEY.IS IT OLD OR NEW (1 OR 2)";Q$
1760 IFQ$="1"ORQ$="2"THENGOTO1770ELSEGOTO1750
1770 IFQ$="1"THENA$(9)="ROMANTIC"ELSEA$(9)="PRACTICAL"
1780 CLS:PRINT:PRINT
1790 PRINT"YOU FIND LIFE ";A$(1)
1800 PRINT"YOU ARE ";A$(2);" YOUR WAY THROUGH IT"
1810 PRINT"YOU ARE ";A$(3);" OF YOUR WAY"
1820 PRINT"YOU LOOK ";A$(4);" ON PEOPLE"
1830 PRINT"YOU ARE ";A$(5);" WITH PEOPLE"
1840 PRINT"YOU ARE A ";A$(6);" PERSON"
1850 PRINT"YOU HAVE ";A$(7);" THOUGHTS"
1860 PRINT"YOU TEND TO BE AN ";A$(8);" PERSON WITH A     ";A$(9);" OUTLOOK ON LIFE"
1870 PRINT:PRINT:PRINT
1880 PRINT" THE RESULTS MAY NOT SEEM RIGHT"
1890 PRINT"BUT ARE THEY ?"
1900 PRINT
1910 PRINT"PRESS ANY KEY TO RUN AGAIN"
1920 FORR=1TO10:A$=INKEY$:NEXT
1930 A$=INKEY$:IFA$=""THENGOTO1930
1940 RUN
```

Scattergraph

Summary

This program, unlike the bar-chart program, is used for generating a graph where the x and y co-ordinates do not need to be related, and points can therefore take any places on the graph rather than being stuck to constant increments on the x-axis.

The graph is automatically scaled to the largest points, and altering incorrect points is made simple.

Once you have the graph you may do what you like with it, perhaps use the screen dump routine to make a permanent copy. This is really a utility which can be included in other programs, perhaps as part of a large statistical packet including lines of regression, or screen dumps.

Program Commentary

40-70   Set up the screen and dimension arrays X,Y for up to 800 values. Sets function key 1 to end input.

80-260   Display title and aim.

270-420   Instructions for the program.

430-620   Data input section. Input is in the format point no., X,Y to allow maximum versatility.

620-770   Present the options to the user (edit, draw, restart) and redirect control to the relevant subroutine.

780-910   Subroutine to edit values previously input.

920-1000   Calculate the X and Y scaling factors for the display.

1010-1030 Input the title for the graph.

1040-1190 Display the graph. The title is plotted using the command OPEN"GRP:", to open the graphics screen as an output channel, and then printing to this channel.

1200-1400 Allow you to list the variables to printer, or return to line 620.

```
10 REM
20 REM SCATTER GRAPH
30 REM
40 SCREEN 0
50 KEYOFF
60 KEY1,"-9,-9,-9"+CHR$(13)
70 DIMX(800),Y(800),P(800)
80 COLOR 15,4,4
90 PRINT:PRINT:PRINT
100 PRINT"   SCATTER GRAPH "
110 PRINT"   ------------"
120 PRINT
130 PRINT" THIS PROGRAM ALLOWS YOU TO ENTER UP "
140 PRINT" TO 800 VALUES AND THEN "
150 PRINT" IT SCALES AND PLOTS THE POINTS."
160 PRINT
170 PRINT" YOU ARE THEN GIVEN THE OPTION TO"
180 PRINT" CHANGE OR DELETE POINTS"
190 PRINT" AND THEN REDRAW."
200 PRINT
210 PRINT
220 PRINT"   PRESS ANY KEY TO CONTINUE."
230 FORR=1 TO 50
240 A$=INKEY$
250 NEXT
260 A$=INKEY$:IFA$="" THEN GOTO 260
270 CLS:PRINT
280 PRINT"   POINT ENTRY"
290 PRINT
300 PRINT" ENTER POINTS IN THE FORMAT ;"
310 PRINT
320 PRINT"   POINT NUMBER , X VALUE , Y VALUE"
330 PRINT
340 PRINT" IF A POINT HAS BEEN SET PREVIOUSLY"
350 PRINT" AN AUDIBLE WARNING WILL BE GIVEN"
360 PRINT" AND YOU WILL BE ASKED TO VERIFY"
370 PRINT" THAT POINT."
380 PRINT
390 PRINT" PLEASE NOTE...ALL VALUES MUST BE POSITIVE."
400 PRINT
410 PRINT"  TO TERMINATE ENTRY PRESS"
420 PRINT"   FUNCTION KEY 1 "
430 PRINT
440 INPUTN,X,Y
450 IFN=-9 AND X=-9 AND Y=-9 THEN GOTO 630
460 IF N<1 OR N>500 THENPRINT" NUMBER OUT OF RANGE":BEEP:GOTO 440
470 IF X<0 THENPRINT" X VALUE OUT OF RANGE":BEEP:GOTO 440
480 IF Y<0 THENPRINT" Y VALUE OUT OF RANGE":BEEP:GOTO 440
490 IFX(N)<>0 THEN GOSUB 540
500 X(N)=X:Y(N)=Y
510 IF N>MN THEN MN=N
520 P(N)=1
530 GOTO 440
540 PLAY"L32O3V9 GA"
```

```
550 PRINT" THIS NUMBER HAS BEEN USED BEFORE."
560 PRINT" THE NUMBER ";N
570 PRINT" CONTAINS  X ";X(N)
580 PRINT"       AND  Y ";Y(N)
590 INPUT" DO YOU WISH TO ENTER THE NEW VALUE   (YES or NO)";YN$
600 IF YN$="YES" THEN GOTO 500
610 IF YN$="NO" THEN GOTO 440
620 BEEP:GOTO 590
630 SCREEN 0
640 PRINT:PRINT
650 PRINT"  1   CHECK POINTS"
660 PRINT"  2   DRAW GRAPH  "
670 PRINT"  3   RESTART"
680 PRINT
690 PRINT" WHICH OPTION ?"
700 FORR=1 TO 50
710 A$=INKEY$
720 NEXT
730 A$=INKEY$:IF A$="" THEN GOTO 730
740 IF A$="3" THEN RUN
750 IF A$="1" THEN GOTO 780
760 IF A$="2" THEN GOTO 920
770 GOTO 730
780 FORR=1 TO MN
790 IFP(R)=0 THEN GOTO 900
800 PRINTR;X(R);Y(R)
810 PRINT" EDIT,DELETE,EXIT OR NEXT (1,2,3,4) ?"
820 A$=INKEY$:IF A$="" THEN GOTO 820
830 IF A$="3" THEN GOTO 630
840 IF A$="2" THEN P(R)=0:X(R)=0:Y(R)=0
850 IF A$<>"1" THEN GOTO 900
860 INPUT " X VALUE , Y VALUE ";X,Y
870 IF X<0 THENPRINT" X VALUE TOO SMALL":BEEP:GOTO 860
880 IF Y<0 THENPRINT" Y VALUE TOO SMALL":BEEP:GOTO 860
890 X(R)=X:Y(R)=Y
900 NEXT
910 GOTO 630
920 CLS:PRINT:PRINT:PRINT:PRINT:PRINT
930 PRINT"   CALCULATIONS IN PROGRESS"
940 FORR=1 TO MN
950 IFP(R)=0 THEN GOTO 980
960 IFX(R)>MX THEN MX=X(R)
970 IFY(R)>MY THEN MY=Y(R)
980 NEXT
990 SX=250/MX
1000 SY=180/MY
1010 CLS:PRINT:PRINT:PRINT:PRINT
1020 INPUT"TITLE (MAX 15 CHARS)";T$
1030 IF LEN(T$)>15 THEN GOTO 1010
1040 REM PLOT THE POINTS
1050 SCREEN 2
1060 LINE(5,0)-(5,182),1
1070 LINE(5,182)-(257,182),1
1080 FORR=1 TO MN
```

detail (handwritten annotation next to lines 920–930)

```
1090 IF P(R)=] THEN PSET(X(R)*SX,180-(Y(R)*SY)),1
1100 NEXT
1110 BEEP
1120 FORR=1 TO 100
1130 A$=INKEY$
1140 NEXT
1150 BEEP
1160 OPEN"GRP:" AS#1
1170 PSET(10,184),4
1180 PRINT#1,T$+" PRESS ANY KEY"
1190 A$=INKEY$:IF A$="" THEN GOTO 1190
1200 SCREEN 0
1210 PRINT:PRINT
1220 PRINT" 1 OPTIONS MENU"
1230 PRINT" 2 PRINTER LIST OF VARIABLES"
1240 PRINT
1250 PRINT" WHICH OPTION ?"
1260 FORR=1 TO 20
1270 A$=INKEY$
1280 NEXT
1290 A$=INKEY$:IF A$="" THEN GOTO 1290
1300 IF A$="1" THEN GOTO 630
1310 IF A$<>"2" THEN GOTO 1290
1320 LPRINT
1330 LPRINT"  VALUES FOR ";T$
1340 LPRINT
1350 FORR=1 TO MN
1360 IF P(R)=0 THEN GOTO 1380
1370 LPRINTR,X(R),Y(R)
1380 NEXT
1390 LPRINT CHR$(12)
1400 GOTO 630
```

Utility

Summary

This program simulates a digital clock, using the whole screen to display LED type characters, and a status line at the top giving information on the alarm and hourly chime.

When the program is run, it starts couting from 00:00:00 immediately. To set the time hit the ´R´ key. You will then be asked to enter the new time. Similarly, to set the alarm, use the ´S´ key. To turn the alarm on or off use the ´A´ key. This also changes the status line at the top. To turn the hourly chime on or off use the ´H´ button . The program can be ended by pressing the space bar.

This program could easily be adapted for more useful purposes, the large tickover display being ideal for a countdown, or for children learning to count.

Program Commentary

40
:   This sets up the general timer for the clock, using the on interval function of the MSX. On other machines this aspect would be very difficult to program, but here it is trivial.

50-120
:   These lines are the controlling loop of the program. Each time through, the computer checks to see if any of the special functions have been requested, and keeps checking until an interrupt is generated. This occurs when the clock reaches a count of 50 cycles.

140-280
:   Set up screen and variables. Array contains graphics strings for digits 0-9.

290-480
:   This is the time-keeping routine. The method is similar to that used in the clock prcgram, but after each update, two variables are set: X is the x-coordinate of the figure to be drawn, and Z is the actual number, used as the subscript of the array N$().

530-700
:   This contains two similar routines, one for setting the alarm and one for setting the clock. Each turns off the interval interrupt, and the alarm routine makes a record of the time elapsed while the user sets it.

710-810
:   Routine to input the time for the clock or the alarm.

820-900
:   Routine to redraw the clock from scratch.

920-980
:   This prints the alarm time, and beeps if time is right.

990-1040
:   This routine prints or deletes the ´H´ signifying hourly chime.

```
10 REM clock
20 GOSUB 140
30 GOSUB 820
40 ON INTERVAL=50 GOSUB 290
50 A$=INKEY$
60 IF A$="H" OR A$="h" THEN C=-C
70 IF A$="A" OR A$="a" THEN A=-A:LINE(112,10)-(120,18),4,BF
80 IF A$="R" OR A$="r" THEN GOSUB 630
90 IF A$="S" OR A$="s" THEN GOSUB 530
100 IF A$=" " THEN GOTO 130
110 GOSUB 990
120 GOTO50
130 END
140 SCREEN2:COLOR15,4,7
150 OPEN"grp:" AS #1
160 C=-1:A=-1
170 N$(0)="r34d68134u68"
180 N$(1)="br34d68"
190 N$(2)="r34d34134d34r34"
200 N$(3)="r34d34nl34d34134"
210 N$(4)="d34r34nu34d34"
220 N$(5)="nr34d34r34d34134"
230 N$(6)="d68r34u34134"
240 N$(7)="r34d68"
250 N$(8)="r34d68134u34nr34u34"
260 N$(9)="r34d34nd34134u34"
270 S=0:M=0:H=0:AL$="":TI$="":T$=""
280 RETURN
290 S=(S+1)MOD 60:GOSUB 340
300 IF S=0 THEN M=(M+1)MOD 60:GOSUB 390
310 IF S=0 AND M=0 THEN H=(H+1)MOD 13:GOSUB 440:IF C=1 THEN BEEP:BEEP
320 IF A=1 THEN GOSUB 910
330 RETURN
340 Z=S MOD 10
350 X=213
360 GOSUB 490
370 IF Z=0 THEN Z=S\10:X=176:GOSUB490
380 RETURN
390 Z=M MOD 10
400 X=130
410 GOSUB 490
420 IF Z=0 THEN Z=M\10:X=93:GOSUB 490
430 RETURN
440 Z=H MOD 10
450 X=47
460 GOSUB 490
470 IF Z=0 THEN Z=H\10:X=10:GOSUB 490
480 RETURN
490 Y=(Z+9)MOD 10
500 PRESET(X,62):DRAW N$(Y)
510 PRESET(X,62):DRAW"c15 xn$(z);"
520 RETURN
530 INTERVAL OFF
540 TIME=0:A=1
```

```
550 H$="alarm"
560 GOSUB 710
570 AL$=T$
580 S=(S+TIME\50)MOD60
590 M=(M+TIME\3000)MOD60
600 H=(H+TIME\18000)MOD13
610 GOSUB 820
620 RETURN
630 INTERVAL OFF
640 H$="clock"
650 GOSUB 710
660 S=0
670 M=VAL(RIGHT$(T$,2))
680 H=VAL(LEFT$(T$,2))
690 GOSUB 820
700 RETURN
710 SCREEN 0:KEYOFF
720 WIDTH 36
730 LOCATE 5,3
740 PRINT"Please enter new time for ";H$
750 PRINTTAB(5)"Two digit format e.g. 03"
760 PRINT"Hours":Q1$=INPUT$(2):PRINTQ1$
770 IF VAL(Q1$)<0 OR VAL(Q1$)>12 THEN PRINT"Hours out of range":GOTO 760
780 PRINT"Minutes":Q2$=INPUT$(2):PRINTQ2$
790 IF VAL(Q2$)<0 OR VAL(Q2$)>59 THEN PRINT"Minutes out of range":GOTO 780
800 T$=Q1$+Q2$
810 RETURN
820 SCREEN 2
830 Z=SMOD10:X=213:GOSUB490
840 Z=S\10:X=176:GOSUB490
850 Z=MMOD10:X=130:GOSUB490
860 Z=M\10:X=93:GOSUB490
870 Z=HMOD10:X=47:GOSUB490
880 Z=H\10:X=10:GOSUB490
890 INTERVAL ON
900 RETURN
910 INTERVAL STOP
920 TI$=STR$(H)+STR$(M)
930 IF VAL(TI$)=VAL(AL$) THEN BEEP
940 AO$=LEFT$(AL$,2)+":"+RIGHT$(AL$,2)
950 PRESET(10,10):PRINT#1,"Alarm ";AO$
960 PRESET(112,10):PRINT#1,"A"
970 INTERVAL ON
980 RETURN
990 INTERVAL STOP
1000 PRESET(100,10)
1010 IF C=-1 THEN LINE(100,10)-(108,18),4,BF
1020 IF C=1 THEN PRINT#1,"H"
1030 INTERVAL ON
1040 RETURN
```

Bar-Chart Generator

Summary

This is one of a number of different types of graph program. Each is used for different types of data. This type of graph is useful for comparing values of something against a continually increasing datum - e.g. monthly sales, daily job losses. The program automatically scales both axes for the largest data, and can automatically handle negative numbers. Alternate bars of the graph are drawn filled in for clarity. Also a title is printed at the top if required.

Program Commentary

20-80   These control the program flow via subroutines.

280-380 This is used to input the data, one at a time. The data values are built up in the array DV(), and the maximum and minimum values compared as it goes along.

390-520 This is a routine which allows you to check the data you have entered, and alter any incorrect ones. The data are printed out in groups of 15 at a time, and you are asked if you wish to change any. If so type ´Y´, and you will be led through the alteration routine, at lines 530-610.

620-850 The plotting routine is generalized, to allow for the possibility of negative numbers. A scaling factor is calculated, SF, and the width of each block, WB. The axes are drawn, and small marks at the top and half-way points of the axes. The blocks are then drawn using the MSX graphics command LINE ...,BF to draw and fill boxes. This process would require a number of lines on most other machines. The x-axis is then numbered at each end, and the title centralized (lines 830-840).

```
10  REM Bar-chart generator
20  GOSUB 90
30  GOSUB 280
40  GOSUB 390
50  GOSUB 620
60  QQ$=INPUT$(1)
70  IF QQ$<>" " THEN GOTO 60
80  END
90  MIN=0:MX=0
100 SCREEN0:WIDTH37:KEYOFF
110 LOCATE9,2:PRINT"Bar-Chart Generator"
120 LOCATE 1,5:PRINT"How many data items (1-100)"
130 LOCATE1,6
140 PRINTSTRING$(37," ")
150 LOCATE1,6:INPUT ND$
160 ND=INT(VAL(ND$))
170 IF NOT(ND>0 AND ND<101) THEN GOTO 130
180 DIM DV(ND)
190 LOCATE 1,8:PRINT"title ";STRING$(31," ")
200 LOCATE7,8:INPUT H$
210 IF LEN(H$)>35 THEN PRINT"Too Long":GOTO 190
220 GOSUB 240
230 RETURN
240 LOCATE5,24:PRINT"Press Space Bar To Continue";
250 QQ$=INPUT$(1)
260 IF QQ$<>" "THEN GOTO 250
270 RETURN
280 CLS
290 LOCATE13,2:PRINT"Data Entry"
300 LOCATE1,4:PRINT"Enter data values at prompt"
310 FOR J=1 TO ND
320 PRINT"Data no. ";STRING$(4-INT(LOG(J)/LOG(10)),".");J
330 INPUT DV(J)
340 IF DV(J)>MX THEN MX=DV(J)
350 IF DV(J)<MIN THEN MIN=DV(J)
360 NEXT J
370 GOSUB 240
380 RETURN
390 FOR I=0 TO (ND-1)\15
400 CLS
410 LOCATE11,1:PRINT"Data Validation"
420 LOCATE4,3:PRINT"Data no.";TAB(15)"Value"
430 NJ=15:IF (ND-I*15)<NJ THEN NJ=(ND-I*15)
440 FOR J=1 TO NJ:N=I*15+J
450 LOCATE5,J+4:PRINTN;TAB(15);DV(N)
460 NEXT J
470 LOCATE1,20:PRINT"Do you wish to change any values?"
480 ANS$=INPUT$(1)
490 IF ANS$="Y" OR ANS$="y" THEN GOSUB 530:GOTO 480
500 GOSUB 240
510 NEXT I
520 RETURN
530 LOCATE1,21:PRINT"Data no.";STRING$(29," ");
540 LOCATE10,21:INPUT DN$
```

```
550 DN=INT(VAL(DN$))
560 IF NOT(DN>0 AND DN<=ND) THEN GOTO 530
570 LOCATE1,22:INPUT"New value ";DV(DN)
580 IF DV(DN)>MX THEN MX=DV(DN)
590 IF DV(DN)<MIN THEN MIN=DV(DN)
600 FOR J=21 TO 22:LOCATE1,J:PRINTSTRING$(37," ");:NEXT
610 RETURN
620 SF=170/(MX-MIN):XA=10+MX*SF
630 WB=240\ND:C=1
640 SCREEN2:COLOR 15,4,7
650 LINE(10,10)-(10,180)
660 LINE(7,XA)-(250,XA)
670 PSET(7,170):DRAW"r3"
680 PSET(7,10):DRAW"r3"
690 PSET(7,(180+XA)/2):DRAW"r3"
700 PSET(7,(10+XA)/2):DRAW"r3"
710 FOR J=10 TO 250 STEP WB
720 PSET(J,XA+3):DRAW"u3"
730 NEXT J
740 FOR J=1 TO ND
750 X1=(J-1)*WB+10:X2=X1+WB
760 Y1=XA:Y2=XA-(DV(J)*SF)
770 C=-C
780 IF C=-1 THEN LINE(X1,Y1)-(X2,Y2),,BF ELSE LINE(X1,Y1)-(X2,Y2),,B
790 NEXT
800 OPEN"grp:"AS #1
810 PRESET(4,XA+2):PRINT#1,"0"
820 PRESET(WB*ND-10,XA+3):PRINT#1,ND
830 HX=128-(LEN(H$)*4)
840 PRESET (HX,1):PRINT#1,H$
850 RETURN
```

Clock

Summary

Not a repeat of the other clock program, but another subroutine for adding to your own masterpieces. This is designed to help keep track of the length of time anything has been running, and displays this time in a simple HH,MM,SS format anywhere on the screen.

This is another of those effort-saving programs that allow you to concentrate on writing good programs, well structured and imaginative, because the boring, mundane bit has already been done for you (see for example the sort programs).

Program Commentary

40-50   Set the internal interrupt generator to send control to the subroutine which updates TI$, this being, naturally, the string which stores the time.

50-70   Set H,M, and S (hours, minutes and seconds) to their starting values.

110-180   These update the clock, print in the top left hand corner then reset the cursor to its original position, using the locations &hF3DD and &hF3DC. This use of memory locations rather than variables makes the program easy to slot into other programs as a routine without too much rewriting of either.

```
10 REM
20 REM CLOCK
30 CLS
40 ON INTERVAL=50 GOSUB 110
50 H=0:REM HOURS
60 M=0:REM MINUTES
70 S=0:REM SECONDS
80 INTERVAL ON
90 REM
100 GOTO 190
110 S=S+1:IFS>59THENS=0:M=M+1
120 IF M>59 THENM=0:H=H+1
130 IF H>23 THENH=0
140 TI$=STR$(H)+STR$(M)+STR$(S)+"   "
150 Y=PEEK(&HF3DC):X=PEEK(&HF3DD)
160 LOCATE0,0:PRINTTI$
170 LOCATEX-1,Y-1
180 RETURN
190 REM YOUR PROGRAM STARTS HERE
```

Intelligence Test

Utility

Summary

This program is <u>not</u> an IQ test. It is merely a set of questions, mostly strung together at random by myself, as a general test of intelligence. An IQ test is standardized over a large sample. Having said that, it doesn't stop you taking a sample of friends and testing them.

The program is simple to follow - you have fifteen minutes to answer the twenty questions, in order. If you get a question right you score five points. If you need help, press the return key before any other. This will print a hint for all but the first question. Be warned - some of the hints are as cryptic as the questions! When you have finished, you are asked if you want to repeat any questions. Answer 'Y' and you are asked the question number. NB All answers to the test should be in lower case.

The question number appears in the top left corner, the time elapsed in minutes in the top right. Once last word - the test is very tricky, so do not be disappointed with a low mark.

Program Commentary

40   Sets interrupt generator for fifteen minutes from now.

60-110   Loop through the twenty questions in order.

120-170   Controls repeating questions and final reckoning.

300-330   Set up answers array, score array.

360-3000   The questions - each produces the screen display and sets variables - M$ (a message), H$ (hint), L (length of answer).

3010-3130 Get answer, return to next question or repeat section as appropriate.

3140-3160 Print hint, reduce score.

3170-3330 Repeat questions ? Redirects control as needed.

3340-3470 Print score and correct answers.

```
10 REM Intelligence test
20 GOSUB 180
30 TIME=0:RF=0
40 ON INTERVAL=(15*3000) GOSUB 160
50 INTERVAL ON
60 FOR I%=1 TO 20
70 ON I% GOSUB 360,560,680,730,840,920,1040,1110,1180,1330
80 IF I%<=10 THEN GOTO 100
90 ONI%-10GOSUB1580,1780,1900,2090,2230,2410,2520,2740,2810,2880
100 GOSUB 3010
110 NEXT
120 RF=1
130 GOSUB 3170
140 IF F=1 THEN 160
150 GOTO130
160 GOSUB 3340
170 END
180 SCREEN0:WIDTH37:COLOR 15,4,7
190 DIM A$(20),S(20),AN$(20)
200 RESTORE 340
210 LOCATE10,1:PRINT"Intelligence Test"
220 PRINT:PRINT:PRINT"This program is designed as a simple"
230 PRINT"measure of your intelligence. It is"
240 PRINT"not an IQ test, as the results are"
250 PRINT"not standardized. You can of course"
260 PRINT"test it amongst your friends."
270 PRINT:PRINT:PRINT"Please enter your name"
280 INPUT NA$
290 NA$=LEFT$(NA$,16)
300 FOR J%=1 TO 20
310 READ A$(J%)
320 S(J%)=5
330 NEXT
340 DATA 2,69,42,4,8,ion,t,60,124,1,e,15,ever,2,10,u,3,p,b,3
350 RETURN
360 SCREEN 2:OPEN"grp:" AS #1
370 RESTORE 380
380 DATA15,0,0,15,30,30,0,30,30,0,15,15,30,0,15,30,15,15,0,15,0,30,30,0
390 FOR J%=1 TO 3
400 FOR K%=1 TO 3
410 IF J%=3 AND K%=3 THEN 460
420 READ X,Y
430 PRESET (10+J%*40,K%*40-40)
440 D$="c15br=x;d30bu=Y;bl=x;r30"
450 DRAW D$:NEXT:NEXT
460 PRESET(141,91):PRINT#1,"?"
470 FOR J%=1 TO 4
480 READ X,Y
490 PRESET (10+J%*40,120)
500 DRAW D$
510 PRESET(14+J%*40,162):PRINT#1,J%
520 NEXT
530 M$="Enter Number Of Choice"
540 L=1:H$=""
```

```
550 RETURN
560 CLS:CIRCLE(65,70),60:CIRCLE(190,70),60
570 RESTORE 580
580 DATA35,30,1,95,30,3,65,70,5,65,110,35,160,30,8,220,30,2,190,70,1,190,110,?
590 FOR J%=1 TO 8
600 READ X,Y,Z$
610 XI=4:IF J%=4 THEN XI=6
620 CIRCLE(X,Y),8:PRESET(X-XI,Y-4)
630 PRINT#1,Z$
640 NEXT
650 M$="Enter Missing Number"
660 L=2:H$="Squaring the circle"
670 RETURN
680 CLS
690 PRESET (10,60):PRINT#1,"18 21 26 33 ??"
700 M$="Enter next number in the series"
710 L=2:H$="Odd differences, eh?"
720 RETURN
730 CLS:RESTORE 740
740 DATA NJHO,TEPRE,NALA,LHNEE
750 FOR J%=1 TO 4
760 PRESET(20,10+J%*20)
770 READ W$:PRINT#1,W$
780 PRESET(80,10+J%*20)
790 PRINT#1,J%
800 NEXT
810 M$="Enter number of odd one out"
820 L=1:H$="What's 'er name"
830 RETURN
840 CLS
850 PRESET(30,30)
860 PRINT#1,"57 (23) 103"
870 PRESET(30,50)
880 PRINT#1,"81 ( ? ) 97"
890 M$="Enter missing number"
900 L=1:H$="That's only half the difference"
910 RETURN
920 CLS
930 PRESET(30,30)
940 PRINT#1,"CARR (---) IZER"
950 PRESET(10,90)
960 PRINT#1,"You must find a 3 letter word"
970 PRESET(10,100)
980 PRINT#1,"to finish the first word and"
990 PRESET(10,110)
1000 PRINT#1,"start the second."
1010 M$="Enter word"
1020 L=3:H$="Dead easy - electrifying"
1030 RETURN
1040 CLS
1050 PRESET(30,30)
1060 PRINT#1,"M Z P W S -"
1070 M$="Enter next letter in series"
1080 L=1
```

```
1090 H$="Alternating currents"
1100 RETURN
1110 CLS
1120 PRESET(30,130)
1130 PRINT#1,"If ADJOIN = 53"
1140 M$="What does KEEPER equal?"
1150 L=2
1160 H$="IM 22 and YOURE 84"
1170 RETURN
1180 CLS:RESTORE 1190
1190 DATA 0,0,3,7,8,26,15,63,24,?
1200 FOR J%=1 TO 5
1210 READ T$,B$
1220 PSET(-10+40*J%,30)
1230 DRAW"r40d80l40u40nr40u40"
1240 PRESET(4+40*J%,46)
1250 PRINT#1,T$
1260 PRESET(4+40*J%,86)
1270 PRINT#1,B$
1280 NEXT
1290 M$="Enter missing number"
1300 L=3
1310 H$="Squares up and cubes down?"
1320 RETURN
1330 CLS:RESTORE 1340
1340 DATA nl20u20,d20,r20,u20d40,r20
1350 FOR J%=1 TO 3
1360 CIRCLE(J%*60-10,30),20
1370 PSET(J%*60-10,30)
1380 READ D$:DRAW D$
1390 NEXT
1400 S$="bh20r40d40l40u40bf20"
1410 FOR J%=1 TO 2
1420 READ D$
1430 PSET(60*J%-10,80)
1440 DRAW S$
1450 DRAW D$
1460 NEXT
1470 FOR J%=1 TO 4
1480 PSET(J%*60-25,130)
1490 DRAW S$
1500 READ D$:DRAW D$
1510 PRESET(J%*60-41,152)
1520 PRINT#1,J%
1530 NEXT
1540 DATA120,nu20r20,nu20nd20nl20r20,nr20l20
1550 M$="Enter number of missing square"
1560 L=1:H$="Add 1 and 2, what's left"
1570 RETURN
1580 CLS:RESTORE 1590
1590 DATA ?,N,V,?,L,O,P,?
1600 CIRCLE(128,96),65
1610 LINE(128,161)-(128,31)
1620 LINE(63,96)-(193,96)
```

```
1630 LINE(82,142)-(174,50)
1640 LINE(82,50)-(174,142)
1650 FOR J=1 TO 8
1660 READ L$
1670 X=128+(40*COS(6.28*J/8+.39))
1680 Y=96+(40*SIN(6.28*J/8+.39))
1690 PRESET(X,Y)
1700 PRINT#1,L$
1710 NEXT
1720 PRESET(10,161)
1730 PRINT#1,"The ? stands for 1 letter"
1740 M$="Enter the missing letter"
1750 L=1
1760 H$="A letter goes in the answer"
1770 RETURN
1780 CLS:RESTORE 1790
1790 DATA 1,3,5,3,6,12,7,1,?
1800 FOR J%=1 TO 3
1810 FOR K%=1 TO 3
1820 READ N$
1830 PRESET(20+J%*20,60+K%*20)
1840 PRINT#1,N$
1850 NEXT:NEXT
1860 M$="Enter the missing number"
1870 L=2
1880 H$="2 x 1 + 2 = 3 Down"
1890 RETURN
1900 CLS:RESTORE 1910
1910 DATA L,F,CL,N,S
1920 FOR J%=10 TO 90 STEP 20
1930 READ L$
1940 PRESET(10,J%)
1950 PRINT#1,L$;
1960 DRAW"bf4"
1970 DRAW"c15m80,54"
1980 NEXT
1990 DRAW"be4"
2000 PRINT#1,"( - - - - )"
2010 PRESET (10,130)
2020 PRINT#1,"Find a 4 letter word which"
2030 PRESET(10,140)
2040 PRINT#1,"also follows the letters shown"
2050 M$="Enter the word"
2060 L=4
2070 H$="Yesterday, today and for..."
2080 RETURN
2090 CLS:RESTORE 2100
2100 DATA2," ",3," ",1,1,?
2110 LINE(40,100)-(190,100)
2120 FOR X=40 TO 205 STEP 25
2130 READ N$:H=5-5*(X=115)
2140 PSET(X,100)
2150 DRAW"d=h;"
2160 PRESET(X-2,90)
```

```
2170 PRINT#1,N$
2180 NEXT
2190 M$="Enter the missing number"
2200 L=1
2210 H$="On balance, it's tricky"
2220 RETURN
2230 CLS:RESTORE 2240
2240 DATA 9,1,?,19,28,4,3,6
2250 CIRCLE(128,96),65
2260 LINE(128,161)-(128,31)
2270 LINE(63,96)-(193,96)
2280 LINE(82,142)-(174,50)
2290 LINE(82,50)-(174,142)
2300 FOR J=1 TO 8
2310 READ L$
2320 X=128+(40*COS(6.28*J/8+1.17))
2330 Y=96+(40*SIN(6.28*J/8+1.17))
2340 PRESET(X,Y)
2350 PRINT#1,L$
2360 NEXT
2370 M$="Enter the missing number"
2380 L=2
2390 H$="Opposites attract"
2400 RETURN
2410 CLS:RESTORE 2420
2420 DATA "G S O G Q","L L O N Q","G S T Q","S H V H A G","L T R O ?","G S T Q"
2430 FOR Y=40 TO 90 STEP 10
2440 READ L$
2450 PRESET(10,Y)
2460 PRINT#1,L$
2470 NEXT
2480 M$="Enter the missing letter"
2490 L=1
2500 H$="Stand up and thinK"
2510 RETURN
2520 CLS
2530 PRESET(104,40):PRINT#1,"Jeremy"
2540 LINE(128,50)-(128,60)
2550 LINE(40,60)-(216,60)
2560 FOR J=40 TO 216 STEP 88
2570 PSET(J,60):DRAW"d10"
2580 NEXT
2590 PRESET(14,72):PRINT#1,"Charles"
2600 PRESET(118,72):PRINT#1,"Ken"
2610 PRESET(198,72):PRINT#1,"Simon"
2620 PSET(40,82):DRAW"d10"
2630 LINE (24,92)-(56,92)
2640 FOR J=1 TO 5
2650 PSET(16+8*J,92):DRAW"d10"
2660 NEXT
2670 PSET(128,82):DRAW"d10l10ndl0r20dl0"
2680 PSET(216,82):DRAW"d10"
2690 PRESET(212,94):PRINT#1,"?"
2700 M$="Simon is father to how many?"
```

```
2710 L=1
2720 H$="Don´t count the letters"
2730 RETURN
2740 CLS
2750 PRESET(10,50)
2760 PRINT#1,"T Y U I O ?"
2770 M$="Enter the next letter"
2780 L=1
2790 H$="Look down"
2800 RETURN
2810 CLS
2820 PRESET(10,50)
2830 PRINT#1,"Y G B B P ?"
2840 M$="Enter the next letter"
2850 L=1
2860 H$="Left you snookered ?"
2870 RETURN
2880 CLS:RESTORE 2890
2890 DATA KOOTY,KAOSA,GEPNIK,SHOMIRAHI
2900 FOR J=1 TO 4
2910 READ W$
2920 PRESET(10,10+J*20)
2930 PRINT#1,W$
2940 PRESET(100,10+J*20)
2950 PRINT#1,J
2960 NEXT
2970 M$="Enter number of odd one out"
2980 L=1
2990 H$="Go east young man"
3000 RETURN
3010 PRESET(235,0):PRINT#1,INT(TIME/3000)
3020 PRESET(0,0):PRINT#1,"Qu ";I%
3030 AN$=""
3040 PRESET(0,172):PRINT#1,M$
3050 FOR Q=1TO10:Q$=INKEY$:NEXT
3060 FOR J%=1 TO L
3070 IN$=INPUT$(1)
3080 IF J%=1 AND ASC(IN$)=13 THEN GOSUB 3140:GOTO3060
3090 PRINT#1,IN$;
3100 AN$=AN$+IN$
3110 NEXT
3120 AN$(I%)=AN$
3130 IF RF=1 THEN RETURN 3170 ELSE RETURN
3140 PRESET(0,182):PRINT#1,H$;" ";
3150 S(I%)=2
3160 RETURN
3170 SCREEN0:WIDTH37:F=0
3180 LOCATE 1,5
3190 PRINT"Do you wish to repeat any questions?"
3200 AN$=INPUT$(1)
3210 IF AN$="N" OR AN$="n" THEN F=1:RETURN
3220 IF AN$<>"Y" AND AN$<>"y" THEN GOTO 3200
3230 LOCATE1,7
3240 PRINT"Which number question ?"
```

```
3250  INPUT NO$:NO=INT(VAL(NO$))
3260  IF NO<0 OR NO>20 THEN GOTO 3250
3270  SCREEN2
3280  I%=NO
3290  ON I% GOSUB 360,560,680,730,840,920,1040,1110,1180,1330
3300  IF I%<=10 THEN GOTO 100
3310  ONI%-10GOSUB1580,1780,1900,2090,2230,2410,?520,2740,2810,2880
3320  GOSUB 3010
3330  RETURN
3340  SCREEN0:SC=0:WIDTH 37:KEYOFF
3350  FOR J%=1 TO 20
3360  IF AN$(J%)=A$(J%) THEN SC=SC+S(J%)
3370  NEXT
3380  TT$="Well done "+NA$
3390  LOCATE 18-LEN(TT$)/2,0
3400  PRINTTT$
3410  PRINT"You have scored ";SC;"%"
3420  PRINT"No.";TAB(6);"Answer";TAB(15);"Your Answer"
3430  FOR J=1 TO 20
3440  PRINT:PRINTJ;TAB(6);A$(J);TAB(15);AN$(J);
3450  NEXT
3460  GOTO 3460
3470  RETURN
```

Qu 15
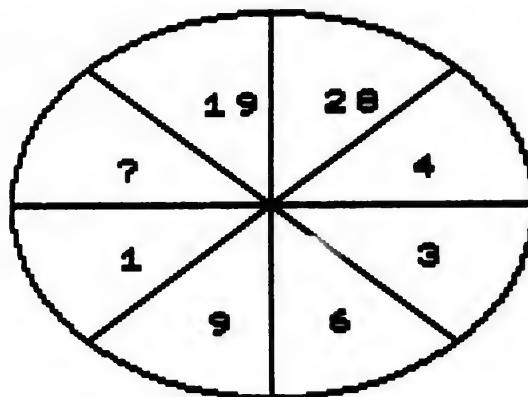
Enter the missing number

Biorhythms

Summary

What can you say about biorhythms that has not been said already ? A fad in the late 70´s, this sort of ´fortune telling´ is ideally suited to a computer, as it involves laborious calculations and plotting of graphs.

The theory of biorhythms is that from your birth, three cycles govern your life. They follow sinusoidal patterns with different periods. The physical cycle takes 23 days, emotional 28, and intellectual 33. Where a cycle is at a low point, you may be prone to accidents, or errors in judgement. At high points good fortune and correct decisions are more likely. Crossovers are critical points.

Some people take these seriously, others light heartedly - whichever way, do not let it spoil your day.

Program Commentary

70-110   Set up data for days in a month.

130-190   Call the instructions.

200-370   Input birthdate, with error trapping.

380-530   Input date for displaying.

540-640   Calculate days elapsed, including leap years.

650-700   Calculate the phase of the three cycles.

710-1070   Display the curves, with titles.

1080-1110 Pause for user to hit a key.

1120-1350 Instructions.

```
10 REM
20 REM BIORHYTHMS
30 REM
40 CLS
50 COLOR 15,4,7
60 KEYOFF
70 DIMDA(12)
80 FORR=1TO12
90 READDA(R)
100 NEXT
110 DATA31,28,31,30,31,30,31,31,30,31,30,31
120 PI=3.142
130 SCREEN0
140 PRINT:PRINT
150 PRINT"    BIORHYTHMS"
160 PRINT"    ----------"
170 PRINT
180 PRINT
190 GOSUB 1150
200 REM THIS IS THE PROGRAM
210 SCREEN 0
220 PRINT:PRINT
230 PRINT"   ENTRY SECTION."
240 PRINT:PRINT:PRINT
250 PRINT" ENTER YOUR BIRTHDATE"
260 PRINT" IN THE FORMAT;"
270 PRINT
280 PRINT"   DAY,MONTH,YEAR"
290 PRINT"EG  14,7,1967"
300 PRINT
310 REM
320 INPUT" BIRTHDATE PLEASE";D,M,Y
330 IFY/4=INT(Y/4)THENDA(2)=29
340 IFY/100=INT(Y/100)ANDY/400<>INT(Y/400)THENDA(2)=28
350 IFY<1ORM<1ORD<1ORM>12THENPRINT"DATE ERROR":PRINT:GOTO320
360 IFY<0ORM<0ORM>12ORD>DA(M)THENPRINT"DATE ERROR":PRINT:GOTO320
370 REM
380 SCREEN 0
390 PRINT:PRINT
400 PRINT"   ENTRY SECTION."
410 PRINT:PRINT:PRINT
420 PRINT" ENTER THEN DATE TO BE"
430 PRINT" DISPLAYED "
440 PRINT" IN THE FORMAT;"
450 PRINT
460 PRINT"   MONTH,YEAR"
470 PRINT"EG  4,1985"
480 PRINT
490 REM
500 INPUT" DATE PLEASE";M2,Y2
510 IFY2<1ORM2<1ORM2>12THENPRINT"DATE ERROR":PRINT:GOTO500
520 IFY2<YOR(Y2=Y1ANDM2<M)THENPRINT"DATE ERROR":PRINT:GOTO500
530 REM
540 CLS:PRINT
```

```
550 PRINT
560 PRINT"     PLEASE WAIT."
570 PRINT"     CALCULATIONS IN PROGRESS."
580 PRINT
590 TDE=(Y2-Y)*365.25
600 TDE=TDE+D
610 FORR=1TOY:DM=DM+DA(R):NEXT
620 IFY/4=INT(Y/4)THENDM=DM+1
630 IFY/100=INT(Y/100)ANDY/400<>INT(Y/400)THENDM=DM-1
640 TDE=TDE+(365-DM)
650 REM PHYSICAL ELAPSED
660 PE=TDE/23
670 REM EMOTIONAL ELAPSED
680 EM=TDE/28
690 REM INTELLECTUAL ELAPSED
700 IN=TDE/33
710 REM DISPLAY
720 SCREEN 2
730 COLOR 15,4,7
740 OPEN"GRP: "AS#1
750 PSET(1,50),4:PRINT#1,"+"
760 PSET(1,130),4:PRINT#1,"-"
770 PSET(25,180),C:PRINT#1,"PHYSICAL."
780 LINE(16,89)-(250,91),1,BF
790 LINE(14,0)-(16,199),1,BF
800 LINE(130,87)-STEP(6,6),1,BF
810 REM
820 PSET(16,90+70*SIN((R-PE)/(11.5*PI))),1
830 FORR=16TO240STEP8
840 LINE-(R,90+70*SIN((R-PE)/(11.5*PI))),1
850 NEXT
860 PRINT#1,"P"
870 REM
880 LINE(25,175)-(250,200),PEEK(&HF3EA),BF
890 PSET(25,180),0:PRINT#1,"EMOTIONAL"
900 LINE(14,0)-(16,199),1,BF
910 REM
920 PSET(16,90+70*SIN((R-EM)/(14*PI))),1
930 FORR=16TO240STEP8
940 LINE-(R,90+70*SIN((R-EM)/(14*PI))),1
950 NEXT
960 PRINT#1,"E"
970 LINE(25,175)-(250,200),PEEK(&HF3EA),BF
980 PSET(25,180),C:PRINT#1,"INTELLECTUAL."
990 LINE(14,0)-(16,199),1,BF
1000 REM
1010 PSET(16,90+70*SIN((R-IN)/(16.5*PI))),1
1020 FORR=16TO240STEP8
1030 LINE-(R,90+70*SIN((R-IN)/(16.5*PI))),1
1040 NEXT
1050 PRINT#1,"I"
1060 LINE(14,0)-(16,199),1,BF
1070 LINE(25,175)-(250,200),PEEK(&HF3EA),BF
1080 PSET(25,180),C:PRINT#1,"SPACE TO CONTINUE."
```

```
1090 FORR=1TO100
1100 A$=INKEY$
1110 NEXT
1120 REM INTRO
1130 A$=INKEY$:IFA$=""THENGOTO1130
1140 RUN
1150 PRINT"   BIORHYTHMS IS A PROGRAM WHICH"
1160 PRINT"   PREDICTS HOW YOU WILL BE"
1170 PRINT"   'FEELING' OVER A GIVEN PERIOD"
1180 PRINT"   OF TIME."
1190 PRINT
1200 PRINT"   IT CALCULATES THIS FOR 3"
1210 PRINT"   CYCLES;"
1220 PRINT
1230 PRINT"   PHYSICAL"
1240 PRINT"   EMOTIONAL  AND "
1250 PRINT"   INTELLECTUAL."
1260 PRINT
1270 PRINT
1280 PRINT"     PRESS SPACE TO CONTINUE"
1290 FORR=1TO100
1300 A$=INKEY$
1310 NEXT
1320 A$=INKEY$
1330 IFA$=""THEN GOTO 1320
1340 CLS
1350 RETURN
```


PHYSICAL
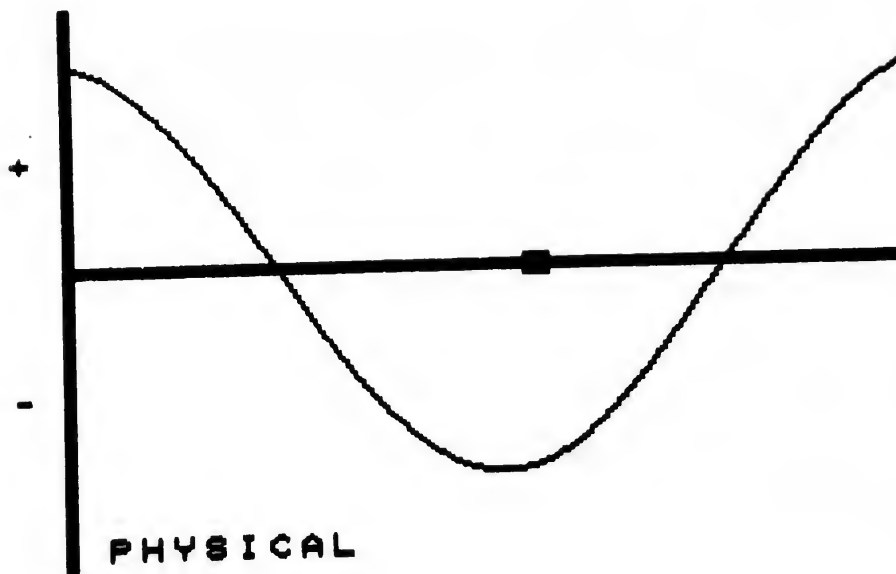
Screen Dump

Summary

This is a very useful subroutine for producing exact copies of graphics programs on dot matrix printers. It was written on a CPA 80, but should work on most Epson printers such as the MX80, RX80 and FX series. The program as it stands should be typed in and added to the end of the program producing the display. It should then be called invisibly, e.g. by pressing the space bar. The program automatically scans for, and ignores, background colours.

The program given will dump a slightly 'squashed' picture, which looks more like the actual screen picture. To get a 'correct' view, with the pixels exactly square, alter the CHR$(4) in line 60 to read CHR$(5).

The program also includes a simple demonstration graphics generator.

Program Commentary

10-50	Graphics demonstration.
60	Sets printer for graphics.
70	Starts scan of Y values.
80	Sets up each line for graphics.
90-120	Scan x-values, print point if not background colour, else print space.
130	Carriage return, line feed.

```
10 SCREEN2
20 PI=ATN(1)*4
30 FORR=0TO2*PISTEP.4
40 LINE(255/2,191/2)-STEP(110*SIN(R),110*COS(R)),1
50 NEXT
60 LPRINTCHR$(27);"3";CHR$(4)
70 FORR=1TO191
80 LPRINTCHR$(27);"K";CHR$(192);CHR$(1);
90 FORT=1TO255
100 A=POINT(T,R):IFA=PEEK(&HF3EA)THENA=0ELSEA=7
110 LPRINTCHR$(A);CHR$(A);
120 NEXT
130 LPRINTCHR$(13)
140 NEXT
```

Convertor

Summary

This is a program to convert metric and imperial values in
length, volume, weight and area. It is fully menu driven – first
you select the main option required, and then a sub-menu is
presented from which you select two items by cursor control. The
first item chosen is the 'from' measure, the second is the 'to'.
E.g. if you first select inches, then yards, you will be
converting values in inches to yards. Selection is made by moving
the cursor up and down with A/Z, and picking an item with the
return key. All will become clear if you run the program.

Program Commentary

30-70   General control loop.

110-240   Print menu, get choice.

290-1280   This contains the subroutines for the sub-menus, and
           they all follow much the same format. Large parts of
           the routines are repetitive, and use subroutines at
           lines 1330, 460, 500, 1590 and 1670.

1330-1470 This prints out the sub-menu from data statements.

1480-1580 Move the cursor, get user's option.

1590-1830 Get number to convert, make the conversion. The method
           used is to fetch a conversion factor from   data
           statements, and use this to convert the 'from' number
           to a standard. The 'to' convertor is then fetched, and
           the conversion completed. This system allows total
           flexibility, for changing any measure to any other if
           the conversion to the standard is known.

1760   A useful one-liner to correct a number to 3 significant
           figures.

```basic
10 REM Metric<>Imperial Convertor
20 GOSUB 80
30 GOSUB 110
40 ON CH GOSUB 290,550,730,1130,1290
50 IF E=1 THEN GOTO 70
60 GOTO30
70 END
80 SCREEN0:WIDTH37
90 KEYOFF:COLOR 15,4,7
100 RETURN
110 CLS:LOCATE16,1:PRINT"Menu"
120 RESTORE 280
130 FOR I%=1 TO 5
140 LOCATE7,3+I%*2
150 READ M$:PRINTI%;") ";M$
160 NEXT
170 LOCATE1,20
180 PRINT"Enter no. of choice ";
190 CH$=INPUT$(1):CH=VAL(CH$)
200 IF CH<1 OR CH>5THEN GOTO 190
210 PRINTCH:PRINT
220 PRINT"Press Space Bar To Continue"
230 GOSUB 250
240 RETURN
250 ZZ$=INPUT$(1)
260 IF ZZ$<>" " THEN GOTO 250
270 RETURN
280 DATA Length,Area,Volume,Weight,End
290 REM length
300 CLS
310 H$="Length"
320 N=7
330 RESTORE 440
340 GOSUB 1330
350 RESTORE 450
360 GOSUB 460
370 RESTORE 450
380 GOSUB 500
390 RESTORE 440
400 GOSUB 1590
410 RESTORE 440
420 GOSUB 1670
430 RETURN
440 DATA Inches,Feet,Yards,Miles,cms,Metres,kms
450 DATA 0.3937,0.0328,0.01094,6.213E-6,1,0.01,0.00001
460 FOR I%=1 TO F
470 READ D
480 NEXT
490 RETURN
500 FOR I%=1 TO T
510 READ M
520 NEXT
530 CO=M/D
540 RETURN
```

```
550 REM area
560 H$="Area"
570 N=9
580 RESTORE 690
590 GOSUB 1330
600 RESTORE 710
610 GOSUB 460
620 RESTORE 710
630 GOSUB 500
640 RESTORE 690
650 GOSUB 1590
660 RESTORE 690
670 GOSUB 1670
680 RETURN
690 DATA Sq. inches,Sq. feet,Sq. yards,Acres,Sq. miles
700 DATA Sq. cms,Sq. metres,Hectares,Sq. kms
710 DATA 0.155,1.076E-3,1.196E-4,2.471E-8,3.861E-11
720 DATA1,1E-4,1E-8,1E-10
730 REM Volume
740 CLS
750 LOCATE 15,1:PRINT"Volume"
760 LOCATE0,3:PRINT"Solid or Liquid volumes(s/l)";
770 AN$=INPUT$(1)
780 IF INSTR("SsLl",AN$)=0 THEN GOTO 770
790 PRINTAN$
800 PRINT:PRINT:PRINT"Press Space Bar To Continue"
810 GOSUB 250
820 IF AN$="L" OR AN$="l" THEN GOTO 960
830 H$="Solid Volume"
840 N=5
850 RESTORE 1090
860 GOSUB 1330
870 RESTORE 1100
880 GOSUB 460
890 RESTORE 1100
900 GOSUB 500
910 RESTORE 1090
920 GOSUB 1590
930 RESTORE 1090
940 GOSUB 1670
950 RETURN
960 H$="Liquid Volume"
970 N=5
980 RESTORE 1110
990 GOSUB 1330
1000 RESTORE 1120
1010 GOSUB 460
1020 RESTORE 1120
1030 GOSUB 500
1040 RESTORE 1110
1050 GOSUB 1590
1060 RESTORE 1110
1070 GOSUB 1670
1080 RETURN
```

```
1090 DATA Cu. inches,Cu. feet,Cu. yards,Cu. cms,Cu. metres
1100 DATA 0.06102,3.531E-5,1.3709E-6,1,1E-6
1110 DATA Fl. oz's,Pints,Gallons,mls,litres
1120 DATA 0.0352,1.76E-3,2.2E-4,1,1E-3
1130 REM Weights
1140 H$="Weight"
1150 N=6
1160 RESTORE 1270
1170 GOSUB 1330
1180 RESTORE 1280
1190 GOSUB 460
1200 RESTORE 1280
1210 GOSUB 500
1220 RESTORE 1270
1230 GOSUB 1590
1240 RESTORE 1270
1250 GOSUB 1670
1260 RETURN
1270 DATA Ounces,Pounds,Tons,Grammes,kgs,Tonnes
1280 DATA 0.03527,2.205E-3,9.842E-7,1,1E-3,1E-6
1290 REM End
1300 CLS
1310 E=1
1320 RETURN
1330 REM Sub-menu
1340 CLS
1350 L=LEN(H$):MID=18-L/2
1360 LOCATE MID,1:PRINTH$
1370 FOR I%=1 TO N
1380 READ M$
1390 LOCATE 4,1+I%*2:PRINTM$;TAB(21)M$
1400 NEXT
1410 LOCATE 1,21:PRINT"A/Z Up/Down        RETURN select"
1420 X=17:Y=3:C$="<":GOSUB 1480:F=TM
1430 X=19:Y=3:C$=">":GOSUB 1480:T=TM
1440 LOCATE 1,23
1450 PRINT"Press Space Bar To Continue";
1460 GOSUB 250
1470 RETURN
1480 LOCATE X,Y:PRINTC$
1490 IN$=INPUT$(1)
1500 LOCATE X,Y:PRINT" "
1510 IF IN$="Z" OR IN$="z" THEN Y=Y+2
1520 IF Y>(1+N*2) THEN Y=3
1530 IF IN$="A" OR IN$="a" THEN Y=Y-2
1540 IF Y<3 THEN Y=1+N*2
1550 IF ASC(IN$)<>13 THEN GOTO 1480
1560 TM=(Y-1)/2
1570 LOCATE X,Y:PRINTC$
1580 RETURN
1590 REM Conversion
1600 CLS
1610 L=LEN(H$):MID=18-L/2
1620 LOCATE MID,1:PRINTH$
```

```
1630 FOR I%=1 TO F
1640 READ F$
1650 NEXT
1660 RETURN
1670 FOR I%=1 TO T
1680 READ T$
1690 NEXT
1700 LOCATE1,3
1710 PRINT"Convert from ";F$;" to ";T$
1720 PRINT"Input ";H$;" in ";F$
1730 INPUT NC$
1740 NC=VAL(NC$)
1750 IF NC=0 THEN PRINT"Numbers only":GOTO1730
1760 CN=NC*CO:TT=10^(INT(LOG(CN)/LOG(10))-2):CN=INT(CN/TT+.5)*TT
1770 PRINT:PRINTNC;" ";F$;" equals ";CN;" ";T$
1780 PRINT:PRINT"Press Space Bar For Menu"
1790 PRINT:PRINT"Press R To Repeat Conversion"
1800 Q$=INPUT$(1)
1810 IF Q$="R" OR Q$="r" THEN GOTO 1720
1820 IF Q$=" " THEN RETURN
1830 GOTO 1800
```

Cassette Database

Summary

This program is intended as much to show how to use the file handling system of the MSX as to provide a useful utility.

The database as it stands is designed to take two FIELDS per RECORD. A number of records is called a FILE or DATABASE. The two fields are called name and telephone number, but could be any two strings. I am not suggesting that anyone would store their telephone numbers on a computer, as retrieval of information is blatantly slower than from a little black book. However the idea can easily be extended to worthwhile projects such as club memberships. Routines from this program, such as the SAVE/LOAD, SORT and SEARCH could then be used almost directly.

When the program is run you are presented with a menu, and you must select an option by typing the number required. At this stage you can only ENTER DATA, LOAD DATA or EXIT the program.

To enter data, type it into the field presented and type return, or space to the end of the line to reach the next field. Hitting return to name and tel. no. of the same record will end the data entry. When searching you may enter any string and the computer will pick out every record containing that string in any position. Pressing the space bar will halt the search until the bar is hit again. When changing a data field, just hitting return when asked for new data leaves the field unchanged.

Program Commentary

20   Clears space for string handling.

60   Redirects control to correct routine.

70-90   End or return to menu.

110   Reserve space for names and tel. nos.

220-230   Input of choice number. Note that the input is a string, as this prevents ´Redo from start´ messages.

540-560   This clears a deleted record to the end of the file.

590-680   Simple bubble sort routine by name.

860   Search routine – tests for presence of SR$ in T$.

970   List up to 10 items or final number if less than 10 left. Note the use of the logical functions true (-1) and false (0) as simple numbers.

1270-1310 Routine to write data to tape.

1400-1470 Reads data from tape knowing the format.

```
10 REM cassette database
20 CLEAR 1000
30 ON ERROR GOTO  1550
40 GOSUB 100
50 GOSUB 150
60 ON CH GOSUB 280,480,590,690,930,1050,1200,1330,1490
70 IF E=1 THEN GOTO 90
80 GOTO 50
90 END
100 SCREEN0:WIDTH 37:COLOR15,4,7
110 DIM NA$(99),TE$(99)
120 N=0
130 KEYOFF
140 RETURN
150 CLS:LOCATE 17,1:PRINT"MENU"
160 RESTORE 270
170 FOR I=1 TO 9
180 READ M$
190 LOCATE 13,2+I*2:PRINTI;"   ";M$
200 NEXT
210 LOCATE 1,22:PRINT" Enter Number Of Choice";
220 CH$=INPUT$(1)
230 CH=VAL(CH$)
240 IF CH<1 OR CH>9 THEN GOTO 220
250 IF N=0 AND (CH<>8 AND CH<>9 AND CH<>1) THEN GOTO 220
260 RETURN
270 DATA Enter data,Delete,Sort,Search,List,Change,Save,Load,End
280 CLS
290 LOCATE1,1:PRINT"No":LOCATE11,1:PRINT"Name":LOCATE26,1:PRINT"Tel. No."
300 FOR I=N+1 TO N+10
310 LOCATE 1,(I-N)*2+3:PRINTI;
320 PRINTTAB(3);:L=19:GOSUB 400
330 NA$(I)=AN$
340 PRINTTAB(24);:L=13:GOSUB 400
350 TE$(I)=AN$
360 IF NA$(I)="" AND TE$(I)="" THEN GOTO 380
370 NEXT
380 N=I-1
390 RETURN
400 AN$=""
410 FOR J=1 TO L
420 A$=INPUT$(1)
430 IF ASC(A$)=13 THEN J=L:GOTO 460
440 PRINTA$;
450 AN$=AN$+A$
460 NEXT
470 RETURN
480 CLS:LOCATE 1,2
490 PRINT"Record No. To Delete"
500 INPUT RN$
510 RN=VAL(RN$)
520 IF RN>N THEN GOTO 580
530 NA$(RN)="":TE$(RN)=""
540 FOR J=RN TO N
```

```
550 SWAP NA$(J),NA$(J+1):SWAP TE$(J),TE$(J+1)
560 NEXT
570 N=N-1
580 RETURN
590 X=0
600 F=0:X=X+1
610 FOR I=1 TO N-X
620 IF NA$(I)<=NA$(I+1) THEN GOTO 660
630 SWAP NA$(I),NA$(I+1)
640 SWAP TE$(I),TE$(I+1)
650 F=1
660 NEXT
670 IF F=1 THEN GOTO 600
680 RETURN
690 CLS
700 LOCATE 1,3
710 PRINT"Do you wish to search for"
720 PRINT"a) A name"
730 PRINT"b) A telephone number"
740 AN$=INPUT$(1)
750 IF AN$<>"a" AND AN$<>"b" THEN 740
760 PRINTAN$
770 PRINT"Enter string to search for"
780 INPUT SR$
790 CLS
800 LOCATE 1,1:PRINT"Searching for ";SR$;" in ";
810 IF AN$="a" THEN PRINT"Names" ELSE PRINT"Tel no.s"
820 LOCATE 1,3:PRINT"No";:PRINTTAB(11)"Name";:PRINTTAB(26);"Tel. No."
830 LOCATE 1,4
840 FOR I=1 TO N
850 IF AN$="a" THEN T$=NA$(I) ELSE T$=TE$(I)
860 IF INSTR(T$,SR$)<>0 THEN PRINTTAB(1);I;TAB(3);NA$(I);TAB(24);TE$(I)
870 IN$=INKEY$:IF IN$=" " THEN GOSUB 920
880 NEXT I
890 LOCATE1,22:PRINT"Press Space Bar To Continue";
900 GOSUB 920
910 RETURN
920 Z$=INKEY$:IF Z$<>" " THEN GOTO 920
930 RETURN
940 FOR I=0 TO (N\10)
950 CLS
960 LOCATE1,1:PRINT"No":LOCATE11,1:PRINT"Name":LOCATE26,1:PRINT"Tel. No."
970 FOR J=1 TO (10*-(I<>(N\10))+(N MOD 10)*-(I=(N\10)))
980 RN=(I*10)+J:Y=(J*2)+1
990 LOCATE1,Y:PRINTRN;TAB(3);NA$(RN);TAB(24);TE$(RN)
1000 NEXT J
1010 LOCATE 1,23
1020 PRINT"Press Space Bar To Continue";
1030 GOSUB 920
1040 NEXT I
1050 RETURN
1060 CLS:LOCATE1,2
1070 PRINT"Record no. to change"
1080 INPUT RN$
1090 RN=VAL(RN$)
```

```
1090 IF RN>=1 AND RN<=N THEN 1110
1100 PRINT"No such record, try again.":GOTO 1070
1110 CLS
1120 PRINT"Record is.."
1130 LOCATE1,6:PRINT"No":LOCATE11,6:PRINT"Name":LOCATE26,6:PRINT"Tel. No."
1140 PRINT:PRINTTAB(1);RN;TAB(3);NA$(RN);TAB(24);TE$(RN)
1150 PRINT:PRINT"Enter new name"
1160 INPUT NA$(RN)
1170 PRINT:PRINT"Enter new number"
1180 INPUT TE$(RN)
1190 RETURN
1200 CLS
1210 LOCATE1,3:PRINT"Saving File - position cassette"
1220 MOTOR ON
1230 LOCATE1,23:PRINT"Press Space When Ready To Save";
1240 PRINT:PRINT:PRINT" Press PLAY and RECORD"
1250 GOSUB 910
1260 MOTOR OFF
1270 OPEN"cas:telno" FOR OUTPUT AS #1
1280 FOR I=1 TO N
1290 PRINT #1,NA$(I);",";TE$(I);",";
1300 NEXT
1310 CLOSE#1
1320 RETURN
1330 CLS
1340 LOCATE1,3:PRINT"Loading File - position tape"
1350 MOTOR ON
1360 LOCATE1,23:PRINT"Press Space When Ready To Load"
1370 GOSUB 910
1380 MOTOR OFF
1390 PRINT:PRINT" Press PLAY on cassette recorder"
1400 OPEN"cas:telno" FOR INPUT AS #1
1410 I=1
1420 IF EOF(1) THEN GOTO 1470
1430 INPUT #1,NA$(I),TE$(I)
1440 I=I+1
1450 LOCATE5,10:PRINT"Record";I
1460 GOTO 1420
1470 N=I-1:CLOSE#1
1480 RETURN
1490 CLS:E=0
1500 LOCATE1,3:PRINT"Have You Saved Your File (Y/N)"
1510 AN$=INPUT$(1)
1520 IF AN$<>"Y" AND AN$<>"y" THEN RETURN
1530 E=1
1540 RETURN
1550 IF ERR=55 THEN RESUME 1470
1560 PRINT"Error ";ERR; "at line ";ERL
1570 STOP
```

Sprite and Character Definers

Summary

These are in fact two separate programs but are so similar that they deserve to be listed together. The programs support both sizes and magnifications of sprites, and all screen modes of characters. Different options are selected by pointing using the cursor keys and selecting with the space bar. The options are:

Colour : Sets the colour of the character/sprite. This is displayed as the border colour.

Set : A cursor is displayed inside the grid which can be moved with the cursor keys. When the space bar is pressed it sets a block of the grid to white. If it is pressed again the block is erased. This can be used to define the sprite pattern. Pressing Q will return the program to the options list.

Code : This is used to encode the design shown in the grid. It is converted first into a series of decimal codes which can be used as CHR$(N) to define the character/sprite. E.g.

SPRITE$(1)=CHR$(23)+CHR$(19)+...

Wipe : This clears the design grid so that unwanted or incorrect designs can quickly be erased.

Save : This saves the character/sprite pattern as a series of characters on tape.

Recall : This reloads a character/sprite from tape.

Exit : Allows you to leave the program. As an aside, it is good practice to include an exit option in menu-based programs such as this, rather than forcing people to break the program. The exit routine then keeps control and can return the computer to its original state, or just give a farewell message.

Program Commentary

Sprites	Charac.	
10-160	10-120	Display option menu.
170-190	130-150	Set up screen for mode selected.
200-290	160-240	Set up variables and arrays.
300-480	250-	Set up defining grid graphics.
490-700	-640	Define the options grid graphics and display with options title inserted.

710-820	650-760	Moves the option list cursor.
830-880	770-820	Fetches user's choice of option and redirects control to it.
890-1100	830-1020	Color option :
890-940	830-880	Displays line of blocks for colour.
950-1040	890-970	Moves cursor.
1050-1100	980-1020	Displays colour number, returns to menu.
1110-1340	1030-1260	Set option :
1110-1140	1030-1060	Displays message and positions cursor.
1150-1260	1070-1180	Moves cursor and sets block in grid if required.
1270-1340	1190-1260	Places/removes a block on the grid.
1350-1610	1270-1570	Code option :
1350-1360	1270-1310	Sets up variables.
1370-1480	1320-1380	Reads defined pattern and displays data generated.
1490-1550	1370	Defines the pattern with the data from the defining grid.
1560-1610	1390-1570	Displays the pattern, returns to menu.
1620-1660	1580-1610	Wipe option :
1620-1660	1580-1610	Clears array, returns to menu.
1670-1720	1620-1690	Save option :
1670-1720	1620-1690	Opens a cassette data stream and saves pattern data.
1730-1800	1700-1870	Recall option :
1730-1750	1700-1810	Open cassette data stream and load data.
1760-1800	1820-1870	Displays pattern, returns to menu.
1810-1880	1890-1950	Exit option :
1810-1880	1890-1950	Stops the program, returns to Basic.
1890-1960	1960-2030	Clears keyboard buffer, message line on screen. (Two subroutines).

```
10 KEYOFF
20 SCREEN1,0,0
30 COLOR15,1,2
40 LOCATE6,6
50 PRINT"SPRITE DEFINER"
60 LOCATE10,10
70 PRINT"WHICH MODE ?"
80 PRINT
90 PRINT"   (0)  8X8  UNMAGNIFIED"
100 PRINT
110 PRINT"   (1)  8X8     MAGNIFIED"
120 PRINT
130 PRINT"   (2) 16X16 UNMAGNIFIED"
140 PRINT
150 PRINT"   (3) 16X16    MAGNIFIED"
160 PRINT
170 INPUTM%
180 IFM%<0ORM%>3THENGOTO170
190 SCREEN1,M%,1
200 LOCATE8,10
210 PRINT"PLEASE WAIT"
220 IFM%<2THEN SIZE%=8 ELSE SIZE%=16
230 DIMG(SIZE%,SIZE%)
240 FORX=1TOSIZE%
250 FORY%=1TOSIZE%
260 G(X%,Y%)=0
270 NEXT:NEXT
280 COLOUR=15:CY%=3
290 GY%=3:GX%=3
300 RESTORE
310 A$=CHR$(1)+"X"
320 B$=CHR$(1)+"T"
330 C$=CHR$(1)+"Z"
340 FORA%=1TOSIZE%
350 A$=A$+CHR$(1)+"R"
360 B$=B$+" "
370 C$=C$+CHR$(1)+"Q"
380 NEXTA%
390 A$=A$+CHR$(1)+"Y"
400 B$=B$+CHR$(1)+"S"
410 C$=C$+CHR$(1)+"["
420 CLS:LOCATE2,2
430 PRINTA$
440 FORA%=1TOSIZE%
450 LOCATE2,A%+2
460 PRINTB$:NEXT
470 LOCATE2,A%+2
480 PRINTC$
490 D$=CHR$(1)+"X"
500 E$=CHR$(1)+"T"
510 F$=CHR$(1)+"Z"
520 FORA%=1TO6:D$=D$+"-"
530 E$=E$+"-":F$=F$+"-"
540 NEXT
```

```
550 D$=D$+CHR$(1)+"Y"
560 E$=E$+CHR$(1)+"S"
570 F$=F$+CHR$(1)+"["
580 LOCATE21,2:PRINTD$
590 FORY%=3TO15STEP2
600 LOCATE21,Y%:READG$
610 PRINTCHR$(1);"V";
620 PRINTG$;CHR$(1);"V"
630 LOCATE21,Y%+1
640 IFY%<15THENPRINTE$ELSEPRINTF$
650 NEXTY%
660 LOCATE8,0
670 PRINT"SPRITE DEFINER"
680 DATA"COLOUR"," SET  "," CODE "
690 DATA" WIPE "," SAVE ","RECALL"
700 DATA" EXIT "
710 LOCATE20,CY%:PRINT">";
720 LOCATE0,20
730 PRINT"PRESS SPACE TO SELECT"
740 K$=INKEY$
750 IFK$=""THENGOTO740
760 K%=ASC(K$):LOCATE20,CY%
770 IFK%=30ANDCY%>3THENPRINT" ":CY%=CY%-2:GOTO 810
780 IFK%=31ANDCY%<14THENPRINT" ":CY%=CY%+2:GOTO 810
790 IFK%=32THENGOTO830
800 GOTO740
810 LOCATE20,CY%
820 PRINT">";:GOTO710
830 C%=(CY%-1)/2
840 LOCATE0,20:PRINT"                    "
850 ONC%GOSUB890,1110,1350,1620,1670,1730,1810
860 GOSUB1890
870 IFC%=4THENGOTO300
880 GOTO710
890 FORA%=4TO19
900 LOCATEA%,20
910 PRINT"O";:NEXT
920 DX%=4:COLOR15
930 LOCATEDX%,21
940 PRINT"^";:COLOR15,,DX%-4
950 K$=INKEY$
960 IFK$=""THENGOTO950
970 K%=ASC(K$)
980 LOCATEDX%,21
990 IFK%=29ANDDX%>4THENPRINT" ":DX%=DX%-1:GOTO1030
1000 IFK%=28ANDDX%<19THENPRINT" ":DX%=DX%+1:GOTO1030
1010 IFK%=32THENGOTO1050
1020 GOTO950
1030 LOCATEDX%,21:PRINT"^";
1040 COLOR15,,DX%-4:GOTO950
1050 COLOUR=DX%-4:LOCATE4,20
1060 PRINT"COLOUR No.=";COLOUR;"   "
1070 LOCATEDX%,21:PRINT" "
1080 K$=INKEY$
```

```
1090 IFK$=""THENGOTO1080
1100 RETURN
1110 LOCATE0,20
1120 PRINT"PRESS ´Q´ TO QUIT"
1130 LOCATEGX%,GY%
1140 PRINT"+";
1150 K$=INKEY$
1160 IFK$=""THENGOTO1110
1170 K%=ASC(K$)
1180 COLOR15,,COLOUR
1190 OX%=GX%:OY%=GY%
1200 IFK%=29ANDGX%>3THENGX%=GX%-1:GOTO1270
1210 IFK%=28ANDGX%<SIZE%+2THENGX%=GX%+1:GOTO1270
1220 IFK%=30ANDGY%>3THENGY%=GY%-1:GOTO1270
1230 IFK%=31ANDGY%<SIZE%+2THENGY%=GY%+1:GOTO1270
1240 IFK%=32THENGOTO1310
1250 IFK%=81THENRETURN
1260 GOTO1110
1270 LOCATEOX%,OY%
1280 IFG(OX%-3,OY%-3)=1THENPRINTCHR$(219)ELSEPRINT" "
1290 LOCATEGX%,GY%
1300 PRINT"+":GOTO1110
1310 G(GX%-3,GY%-3)=(G(GX%-3,GY%-3)-1)*-1
1320 IFG(GX%-3,GY%-3)=1THENCH=219ELSECH=32
1330 LOCATEGX%,GY%:PRINTCHR$(CH)
1340 COLOR15,,COLOUR:GOTO1110
1350 CODE$="":LOCATE0,20
1360 P%=0:Q%=7:R=0
1370 FORA%=0TOSIZE%-1
1380 BI$="&b"
1390 FORB%=P%TOQ%
1400 BI$=BI$+CHR$(G(B%,A%)+48)
1410 NEXTB%:BI%=VAL(BI$)
1420 PRINTBI%;"/";
1430 CODE$=CODE$+CHR$(BI%)
1440 X=X+1
1450 IFX>=4THENGOSUB1910
1460 NEXTA%
1470 IFA%=16ANDF=0THENP%=8:Q%=SIZE%-1:F=1:GOTO1370
1480 F=0
1490 K$=INKEY$
1500 IFK$=""THENGOTO 1490
1510 LOCATE0,20
1520 PRINT"
1530 LOCATE2,20
1540 INPUT"SPRITE NUMBER ";SN%
1550 SPRITE$(SN%)=CODE$
1560 LOCATE2,20
1570 PRINT"SPRITE ";SN%;"=>           "
1580 PUTSPRITE1,(200,150),COLOUR,SN%
1590 K$=INKEY$
1600 IFK$<>""THENGOTO710
1610 GOTO1590
1620 FORA%=0TOSIZE%-1
```

```
1630 FORB%=0TOSIZE%-1
1640 G(A%,B%)=0:NEXT:NEXT
1650 SPRITE$(SN%)=""
1660 RETURN
1670 LOCATE0,20
1680 INPUT"PRESS RECORD THEN RETURN";A$
1690 OPEN"CAS:DATA"FOROUTPUTAS#1
1700 PRINT#1,CODE$
1710 CLOSE#1
1720 RETURN
1730 OPEN"CAS:DATA"FORINPUTAS#1
1740 INPUT#1,CODE$
1750 CLOSE#1
1760 LOCATE0,20
1770 INPUT"SPRITE NUMBER";SN
1780 SPRITE$(SN)=CODE$
1790 PUTSPRITE1,(200,150),COLCUR,SN
1800 RETURN
1810 LOCATE2,20
1820 PRINT"ARE YOU SURE ?
1830 K$=INKEY$
1840 IFK$=""THENGOTO 1830
1850 K%=ASC(K$)
1860 IFK%=89THEN END
1870 IFK%=78THENRETURN
1880 GOTO 1830
1890 FORA=1TO100
1900 K$=INKEY$:NEXTA:RETURN
1910 K$=INKEY$
1920 IFK$=""THENGOTO 1910
1930 LOCATE0,20:X=0
1940 PRINT"
1950 LOCATE0,20
1960 RETURN
```

```
10 KEYOFF
20 SCREEN1,0,0
30 COLOR15,1,2
40 LOCATE6,6
50 PRINT"CHARACTER DEFINER"
60 LOCATE10,10
70 PRINT"WHICH SCREEN ?"
80 PRINT:PRINT:PRINT
90 PRINT"  (0)  6X8  CHARACTERS"
100 PRINT
110 PRINT"  (1)  8X8  CHARACTERS"
120 PRINT
130 INPUTM%
140 IFM%<0ORM%>1THENGOTO130
150 SCREEN1,0,1
160 IFM%=0THEN SIZE%=6 ELSE SIZE%=8
170 DIMG(8,8)
180 FORX=1TO8
190 FORY%=1TO8
200 G(X%,Y%)=0
210 NEXT:NEXT
220 COLOUR=15:CY%=6
230 GY%=6:GX%=6
240 RESTORE
250 A$=CHR$(1)+"X"
260 B$=CHR$(1)+"T"
270 C$=CHR$(1)+"Z"
280 FORA%=1TOSIZE%
290 A$=A$+CHR$(1)+"R"
300 B$=B$+" "
310 C$=C$+CHR$(1)+"Q"
320 NEXTA%
330 A$=A$+CHR$(1)+"Y"
340 B$=B$+CHR$(1)+"S"
350 C$=C$+CHR$(1)+"["
360 D$=CHR$(1)+"X"
370 E$=CHR$(1)+"T"
380 F$=CHR$(1)+"Z"
390 FORA%=1TO6:D$=D$+"-"
400 E$=E$+"-":F$=F$+"-"
410 NEXT
420 D$=D$+CHR$(1)+"Y"
430 E$=E$+CHR$(1)+"S"
440 F$=F$+CHR$(1)+"["
450 CLS:LOCATE5,5
460 PRINTA$
470 FORA%=1TO8
480 LOCATE5,A%+5
490 PRINTB$:NEXT
500 LOCATE5,A%+5
510 PRINTC$
520 LOCATE21,5:PRINTD$
530 FORY%=6TO18STEP2
540 LOCATE21,Y%:READG$
```

```
550 PRINTCHR$(1);"V";
560 PRINTG$;CHR$(1);"V"
570 LOCATE21,Y%+1
580 IFY%<18THENPRINTE$ELSEPRINTF$
590 NEXTY%
600 LOCATE6,0
610 PRINT"CHARACTER DEFINER"
620 DATA"COLOUR"," SET  "," CODE "
630 DATA" WIPE "," SAVE ","RECALL"
640 DATA" EXIT "
650 LOCATE20,CY%:PRINT">";
660 LOCATE0,20
670 PRINT"PRESS SPACE TO SELECT"
680 K$=INKEY$
690 IFK$=""THENGOTO680
700 K%=ASC(K$):LOCATE20,CY%
710 IFK%=30ANDCY%>6THENPRINT" ":CY%=CY%-2:GOTO 750
720 IFK%=31ANDCY%<17THENPRINT" ":CY%=CY%+2:GOTO 750
730 IFK%=32THENGOTO770
740 GOTO680
750 LOCATE20,CY%
760 PRINT">";:GOTO650
770 C%=(CY%-4)/2
780 LOCATE0,20:PRINT"                      "
790 ONC%GOSUB830,1030,1270,1580,1620,1700,1880
800 GOSUB1960
810 IFC%=4THENGOTO240
820 GOTO650
830 FORA%=4TO19
840 LOCATEA%,20
850 PRINTCHR$(219);:NEXT
860 DX%=4:COLOR15
870 LOCATEDX%,21
880 PRINT"^";:COLOR15,,DX%-4
890 K$=INKEY$
900 IFK$=""THENGOTO890
910 K%=ASC(K$)
920 LOCATEDX%,21
930 IFK%=29ANDDX%>4THENPRINT" ":DX%=DX%-1:GOTO970
940 IFK%=28ANDDX%<19THENPRINT" ":DX%=DX%+1:GOTO970
950 IFK%=32THENGOTO990
960 GOTO890
970 LOCATEDX%,21:PRINT"^";
980 COLOR15,,DX%-4:GOTO890
990 COLOUR=DX%-4:LOCATE4,20
1000 PRINT"           "
1010 LOCATEDX%,21:PRINT" "
1020 RETURN
1030 LOCATE0,20
1040 PRINT"PRESS 'Q' TO QUIT"
1050 LOCATEGX%,GY%
1060 PRINT"+";
1070 K$=INKEY$
1080 IFK$=""THENGOTO1030
```

```
1090 K%=ASC(K$)
1100 COLOR15,, COLOUR
1110 OX%=GX%:OY%=GY%
1120 IFK%=29ANDGX%>6THENGX%=GX%-1:GOTO1190
1130 IFK%=28ANDGX%<SIZE%+5THENGX%=GX%+1:GOTO1190
1140 IFK%=30ANDGY%>6THENGY%=GY%-1:GOTO1190
1150 IFK%=31ANDGY%<13THENGY%=GY%+1:GOTO1190
1160 IFK%=32THENGOTO1230
1170 IFK%=81THENRETURN
1180 GOTO1030
1190 LOCATEOX%,OY%
1200 IFG(OX%-6,OY%-6)=1THENPRINTCHR$(219)ELSEPRINT" "
1210 LOCATEGX%,GY%
1220 PRINT"+":GOTO1030
1230 G(GX%-6,GY%-6)=(G(GX%-6,GY%-6)-1)*-1
1240 IFG(GX%-6,GY%-6)=1THENCH=219ELSECH=32
1250 LOCATEGX%,GY%:PRINTCHR$(CH)
1260 COLOR15,,COLOUR:GOTO1030
1270 LOCATE1,20
1280 PRINT"                    "
1290 LOCATE1,20
1300 INPUT"CHARACTER No.";N%
1310 IFN%<33ORN%>255THENGOTO1270
1320 FORA%=0TO7
1330 BI$="&b"
1340 FORB%=0TO7
1350 BI$=BI$+CHR$(G(B%,A%)+48)
1360 NEXTB%:C(A%)=VAL(BI$)
1370 VPOKE(N%*8)+A%,C(A%)
1380 NEXTA%
1390 LOCATE0,20
1400 PRINT"                    "
1410 LOCATE0,20
1420 PRINTC(0);C(1);C(2);C(3)
1430 LOCATE0,21
1440 PRINTC(4);C(5);C(6);C(7)
1450 K$=INKEY$
1460 IFK$=""THENGOTO 1450
1470 LOCATE0,20
1480 PRINT"                    "
1490 LOCATE0,21
1500 PRINT"                    "
1510 LOCATE1,20
1520 COLORCOLOUR
1530 PRINT"CHARACTER No. ";N%;"=";CHR$(N%)
1540 K$=INKEY$
1550 IFK$=""THENGOTO1540
1560 COLOR15
1570 GOTO650
1580 FORA%=0TO7
1590 FORB%=0TO7
1600 G(A%,B%)=0:NEXT:NEXT
1610 RETURN
1620 LOCATE0,20
```

```
1620 LOCATE0,20
1630 INPUT"PRESS RECORD THEN RETURN";A$
1640 LOCATE0,20
1650 PRINT"                                "
1660 OPEN"CAS:DATA"FOROUTPUTAS#1
1670 PRINT#1,C(0),C(1),C(2),C(3),C(4),C(5),C(6),C(7)
1680 CLOSE#1
1690 RETURN
1700 LOCATE0,20
1710 PRINT"ARE YOU SURE ?"
1720 K$=INKEY$
1730 IFK$=""THENGOTO1720
1740 IFK$="Y"THENGOTO 1770
1750 IFK$="N"THENRETURN
1760 GOTO1720
1770 LOCATE0,20
1780 PRINT"                         "
1790 OPEN"CAS:DATA"FORINPUTAS#1
1800 INPUT#1,C(0),C(1),C(2),C(3),C(4),C(5),C(6),C(7)
1810 CLOSE#1
1820 LOCATE0,20
1830 INPUT"CHARACTER NUMBER";N%
1840 FORA%=0TO7
1850 VPOKE(N%*8)+A%,C(A%)
1860 NEXTA%
1870 GOTO1390
1880 LOCATE2,20
1890 PRINT"ARE YOU SURE ?
1900 K$=INKEY$
1910 IFK$=""THENGOTO 1900
1920 K%=ASC(K$)
1930 IFK%=89THEN END
1940 IFK%=78THENRETURN
1950 GOTO 1900
1960 FORA=1TO100
1970 K$=INKEY$:NEXTA:RETURN
1980 K$=INKEY$
1990 IFK$=""THENGOTO 1980
2000 LOCATE0,20:X=0
2010 PRINT"
2020 LOCATE0,20
2030 RETURN
```

Sketch

Summary

Remember etch-a-sketch ? Computer technology has finally caught up, and using several hundred pounds worth of equipment you can emulate a cheap game ! Seriously though, this program is the basis of a computer aided design (CAD) system, allowing drawing of lines, circles, arcs, boxes, and filling of shapes in the screen 2 (hi-res) mode. You may like to use the screen dump to get 'hard copies' of your designs, or just doodle to your heart's content.

Program Commentary

90-190   Sets up array for sin and cosine.

1250   Generates sprite for cursor.

1280-1330 Move the cursor if required.

1340-1380 Call subroutines for each function - line, circle, etc.

1410-1500 Routine to draw line.

1510-1750 Routine to draw ellipse.

1760-1850 Routine to draw rectangle.

1860-1930 Changes foreground colour.

1940-2020 Routine to fill area.

```
10 REM
20 REM
30 REM     SCREEN 2
40 REM     GRAPHICS DESIGNER - SKETCH PAD
50 REM
60 REM        22/12/1984
70 REM
80 SCREEN 0
90 DIMSI(360),CO(360)
100 CLS
110 PRINT
120 PRINT
130 PRINT"    SKETCH PAD "
140 LOCATE10,10:PRINT" SETTING UP VARIABLES."
150 PI=3.14162
160 A!= .017453292370619#
170 FORR=0 TO 360:SI(R)=SIN(R*A!):CO(R)=COS(R*A!)
180 LOCATE 15,5:PRINTINT(((360-R)/360)*100);" "
190 NEXT
200 SCREEN 0:CLS
210 COLOR 1,4,7
220 KEYOFF
230 BEEP
240 PRINT
250 PRINT
260 PRINT"    SKETCH PAD "
270 PRINT
280 PRINT
290 PRINT" USE THE CURSOR KEYS TO "
300 PRINT"TO SET A COLOUR PRESS"
310 PRINT" THE SPACE BAR THEN
320 PRINT" TYPE THE NUMBER OF THE"
330 PRINT" COLOUR AS IN THE MANUAL
340 PRINT " THEN PRESS RETURN."
350 PRINT
360 PRINT
370 PRINT" PRESS ´A´ FOR INSTRUCTIONS"
380 PRINT" PRESS ´B´ FOR SKETCH PAD"
390 A$=INKEY$:IFA$=""THEN GOTO 390
400 IF A$="A" THEN GOTO 430
410 IF A$="B" THEN GOTO 1200
420 GOTO 390
430 CLS
440 PRINT
450 PRINT
460 PRINT " TO DRAW A LINE BETWEEN TWO POINTS"
470 PRINT " POSITION THE CURSOR"
480 PRINT " PRESS   ´L´ "
490 PRINT " POSITION THE OTHER END"
500 PRINT " THEN PRESS   ´L´ AGAIN."
510 PRINT
520 PRINT " THE LINE WILL BE DRAWN IN"
530 PRINT " THE CURRENT COLOUR."
540 PRINT
```

```
550 PRINT
560 PRINT" PRESS ANY KEY TO CONTINUE "
570 A$=INKEY$:IFA$=""THEN GOTO 570
580 CLS
590 PRINT
600 PRINT
610 PRINT" TO DRAW A CIRCLE "
620 PRINT" POSITION THE CURSOR IN THE CENTER"
630 PRINT" PRESS   'C'"
640 PRINT" SET THE X AXIS"
650 PRINT" PRESS   'C'"
660 PRINT" SET THE Y AXIS"
670 PRINT" THEN PRESS   'C' AGAIN."
680 PRINT
690 PRINT" THE CIRCLE WILL BE DRAWN IN"
700 PRINT" THE CURRENT COLOUR."
710 PRINT
720 PRINT
730 PRINT" PRESS ANY KEY TO CONTINUE "
740 A$=INKEY$:IFA$=""THEN GOTO 740
750 CLS
760 PRINT
770 PRINT
780 PRINT " TO DRAW AN ARC YOU MUST"
790 PRINT " DRAW A CIRCLE AS BEFORE"
800 PRINT " THEN ERASE PARTS OF THE"
810 PRINT " CIRCLE BY DRAWING LINES "
820 PRINT " OVER THE UNWANTED AREA ."
830 PRINT
840 PRINT
850 PRINT" PRESS ANY KEY TO CONTINUE "
860 A$=INKEY$:IFA$=""THEN GOTO 860
870 CLS
880 PRINT
890 PRINT
900 PRINT " TO DRAW A RECTANGLE"
910 PRINT " POSITION THE CURSOR"
920 PRINT " TYPE   'R'"
930 PRINT " POSITION THE CURSOR AT THE OPPOSITE"
940 PRINT"CORNER"
950 PRINT " THEN TYPE   'R' AGAIN."
960 PRINT
970 PRINT
980 PRINT" PRESS ANY KEY TO CONTINUE "
990 A$=INKEY$:IFA$=""THEN GOTO 990
1000 CLS
1010 PRINT
1020 PRINT
1030 PRINT " TO FILL AN AREA WITH A COLOUR"
1040 PRINT " POSITION THE CURSOR IN THE AREA"
1050 PRINT "TO BE FILLED."
1060 PRINT " THEN TYPE   'F'"
1070 PRINT
1080 PRINT " IF THE COLOUR IS WRONG OR"
```

```
1090 PRINT " IT LEAKS SIMPLY REPEAT THE"
1100 PRINT " PROCESS USING THE BACKGROUND"
1110 PRINT " COLOUR."
1120 PRINT
1130 PRINT
1140 PRINT" PRESS ANY KEY TO CONTINUE "
1150 A$=INKEY$:IFA$=""THEN GOTO1150
1160 CLS
1170 PRINT
1180 PRINT
1190 GOTO 200
1200 REM
1210 REM SKETCH PAD
1220 REM
1230 REM
1240 SCREEN 2,0
1250 SPRITE$(1)=CHR$(224)+CHR$(192)+CHR$(160)+CHR$(16)+CHR$(0)+CHR$(0)+CHR$(0)+CHR$(0)
1260 X=100:Y=100:C=1
1270 PUTSPRITE1,(X,Y),1
1280 A$=INKEY$:IFA$=""THEN GOTO1280
1290 IF A$=CHR$(29)ANDX>3THENX=X-1
1300 IF A$=CHR$(28)ANDX<256THENX=X+1
1310 IF A$=CHR$(30)ANDY>0THENY=Y-1
1320 IF A$=CHR$(31)ANDY<192THENY=Y+1
1330 PSET (X,Y),C
1340 IFA$="L"THEN GOTO 1410
1350 IFA$="C"THEN GOTO 1510
1360 IFA$="R"THEN GOTO 1760
1370 IF A$="F"THEN GOTO 1940
1380 IFA$=" "THEN GOTO 1860
1390 GOTO 1270
1400 IF A$=CHR$(31)ANDY<192THENY=Y+1
1410 XS=X:YS=Y
1420 A$=INKEY$:IFA$=""THEN GOTO1420
1430 IF A$=CHR$(29)ANDX>3THENX=X-1
1440 IF A$=CHR$(28)ANDX<256THENX=X+1
1450 IF A$=CHR$(30)ANDY>0THENY=Y-1
1460 IF A$=CHR$(31)ANDY<192THENY=Y+1
1470 PUTSPRITE1,(X,Y),1
1480 IF A$<>"L"THEN GOTO 1420
1490 LINE(XS,YS)-(X,Y),C
1500 GOTO 1270
1510 XS=X:YS=Y
1520 A$=INKEY$:IFA$=""THEN GOTO1520
1530 IF A$=CHR$(29)ANDX>3THENX=X-1
1540 IF A$=CHR$(28)ANDX<256THENX=X+1
1550 PUTSPRITE1,(X,Y),1
1560 IF A$<>"C"THENGOTO 1520
1570 X2=X
1580 X=XS:PUTSPRITE1,(X,Y),1
1590 A$=INKEY$:IFA$=""THEN GOTO1590
1600 IF A$=CHR$(30)ANDY>0THENY=Y-1
1610 IF A$=CHR$(31)ANDX<191THENY=Y+1
1620 PUTSPRITE1,(X,Y),1
```

```
1630 IF A$<>"C"THENGOTO 1590
1640 S=2:P2=90:P3=270:P4=180
1650 FORR=0 TO 88 STEP S
1660 A(1)=(X2-X)*SI(R)+XS:A(2)=(YS-Y)*CO(R)+YS:A(3)=(X2-X)*SI(R+S)+XS
1670 A(4)=(YS-Y)*CO(R+S)+YS:LINE(A(1),A(2))-(A(3),A(4)),C
1680 A(1)=(X2-X)*SI(R+P4)+XS:A(2)=(YS-Y)*CO(R+P4)+YS:A(3)=(X2-X)*SI(R+S+P4)+XS
1690 A(4)=(YS-Y)*CO(R+S+P4)+YS:LINE(A(1),A(2))-(A(3),A(4)),C
1700 A(1)=((X2-X)*SI(R+P2)+XS):A(2)=((YS-Y)*CO(R+P2)+YS):A(3)=((X2-X)*SI(R+S+P2)+XS
1710 A(4)=(YS-Y)*CO(R+S+P2)+YS:LINE(A(1),A(2))-(A(3),A(4)),C
1720 A(1)=(X2-X)*SI(R+P3)+XS:A(2)=(YS-Y)*CO(R+P3)+YS:A(3)=(X2-X)*SI(R+S+P3)+XS
1730 A(4)=(YS-Y)*CO(R+S+P3)+YS:LINE(A(1),A(2))-(A(3),A(4)),C
1740 NEXT
1750 X=XS:Y=YS:GOTO 1270
1760 XS=X:YS=Y
1770 A$=INKEY$:IFA$=""THEN GOTO1770
1780 IF A$=CHR$(29)ANDX>3THENX=X-1
1790 IF A$=CHR$(28)ANDX<256THENX=X+1
1800 IF A$=CHR$(30)ANDY>0THENY=Y-1
1810 IF A$=CHR$(31) ANDY<191THENY=Y+1
1820 PUTSPRITE1,(X,Y),1
1830 IFA$<>"R"THENGOTO1770
1840 LINE(XS,YS)-(X,Y),C,B
1850 GOTO 1270
1860 A$=INKEY$:C$=""
1870 A$=INKEY$:IFA$=""THENGOTO1870
1880 IFA$=CHR$(13)THENC=VAL(C$):GOTO 1920
1890 C$=C$+A$
1900 IFLEN(C$)>2THENC$="":BEEP
1910 GOTO1870
1920 IF C<0 OR C>15 THEN BEEP:C$="":GOTO 1870
1930 GOTO 1270
1940 A$=INKEY$:IFA$=""THEN GOTO1940
1950 IF A$=CHR$(29)ANDX>3THENX=X-1
1960 IF A$=CHR$(28)ANDX<256THENX=X+1
1970 IF A$=CHR$(30)ANDY>0THENY=Y-1
1980 IF A$=CHR$(31)ANDY<192THENY=Y+1
1990 PUTSPRITE1,(X,Y),1
2000 IFA$ <> "F" THEN GOTO 1940
2010 PAINT(X,Y),C,C
2020 GOTO 1270
```

Julian

Summary

Want to know what day your birthday falls on ? This program
provides a general purpose routine to calculate Julian dates –
i.e the day of any year from 1752 onwards. Before 1752 we used
the Gregorian calendar, with which this program is not
compatible. The changeover in calenders involved the loss of 11
days – much to the annoyance of the populace who believed their
lives had been shortened !

Program Commentary

70-150	Read days and month lengths data into arrays.
400-560	Enter and validate the date.
530-560	Tests for leap year.
570-630	Calculates number of days elapsed in year.
640-780	Deliver result to user.
790-850	Control rerunning of program.

```
1U REM
20 REM JULIAN DATE CALCULATOR
30 REM
40 REM
50 Z=0
60 SCREEN 1
70 DATA SUNDAY,MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,SATURDAY
80 DATA 31,28,31,30,31,30,31,31,30,31,30,31
90  FOR R= 1 TO 7
100 READD$(R)
110 NEXT
120 DIMD(12)
130 FORR=1 TO 12
140 READD(R)
150 NEXT
160 T=0
170 KEYOFF
180 CLS:COLOR4,1,1
190 FOR R = 1 TO 3
200 PRINT
210 NEXT
220 PRINT "    JULIAN DATE CALCULATOR"
230 PRINT "    AND DAY CALCULATOR"
240 PRINT
250 PRINT" The Julian date is           calculated by using";
260 PRINT" January  the first of the year chosen ";
270 PRINT"as the base date and countingfrom there."
280 PRINT
290 PRINT" If the year entered is after1752 it is also ";
300 PRINT"possible to  calculate the day."
310 PRINT
320 PRINT "ENTER THE DATE THUS;"
330 PRINT"DAY"
340 PRINT"MONTH"
350 PRINT"YEAR (AD AND IN FULL)"
360 PRINT
370 PRINT "PRESS A KEY TO ENTER DATE"
380 A$=INKEY$:IFA$=""THENGOTO380
390 CLS
400 PRINT
410 INPUT" WHAT DAY (Eg 10)";D
420 IFD<1THENPRINT"THERE IS A MINIMUM OF 1 DAY  IN A MONTH ":BEEP:PRINT:GOTO400
430 IF D>31THENPRINT"THERE IS A MAXIMUM OF 31 DAYSIN A MONTH":BEEP:PRINT:GOTO400
440 CLS
450 IFT=1THENGOTO530
460 PRINT
470 INPUT" WHAT MONTH (Eg 7)";M
480 IFM<1THENPRINT"THERE IS A MINIMUM OF 1 MONTHIN A YEAR":BEEP:PRINT:GOTO460
490 IFM>12THENPRINT"THERE IS A MAXIMUM OF 12    MONTHS IN A YEAR":BEEP:PRINT:GOTO460
500 CLS
510 PRINT
520 INPUT" WHAT YEAR (AD AND IN FULL   Eg 1984)";Y
530 IFY/4=INT(Y/4)THEN D(2)=29
540 IFY/100=INT(Y/100)ANDY/400<>INT(Y/400)THEN D(2)=28
```

```basic
550 IFY<0THENPRINT"AD ONLY":BEEP:PRINT:GOTO510
560 IFD>D(M)THENPRINT"THERE IS A MAXIMUM OF ";D(M);"DAYS IN THAT MONTH.":T=1:GOTO400
570 REM THIS IS THE
580 REM CALCULATION BIT.
590 IF M=1 THEN GOTO 630
600 FORR=1TOM-1
610 YT=YT+D(R)
620 NEXT
630 YT=YT+D
640 CLS
650 PRINT
660 PRINT" THE JULIAN DATE FOR ":PRINTD;"/";M;"/";Y
670 PRINT "IS ";YT;" OF";Y
680 PRINT
690 IFD(2)=29 THEN PRINT"  THE YEAR IS A LEAP YEAR "
700 IF Y<1752 THEN GOTO 790
710 M=M-2:IFM<1THENM=M+12:Y=Y-1
720 C=INT(Y/100)
730 Y=VAL(RIGHT$(STR$(Y),2))
740 A=INT(2.6*M-.19)+D+Y+INT(Y/4)+INT(C/4)-C*2
750 A=INT((A/7-INT(A/7))*7+.1)+1
760 L=A
770 PRINT
780 PRINT "AND THE DAY IS A ";D$(L)
790 REM THIS IS THE RE-RUN
800 REM BIT
810 PRINT
820 PRINT "DO YOU WISH TO FIND ANOTHER  DATE (Y/N) ?"
830 A$=INKEY$:IFA$="Y"ORA$="y"THENRUN
840 IFA$="N"ORA$="n"THENCLS:PRINT:PRINT:END
850 GOTO830
```

Trigonometry

Summary

Back to the schoolbooks - remember SIN, COS and TAN ? This program is designed to take the slog out of solving triangle problems. You are presented with a menu which asks how much information you have about the triangle. If you have any combination of data other than those in the menu then the triangle cannot be solved. If you have more data, you can choose any of the options possible. Once you have chosen, you are asked to enter the data, remembering that all angles are in degrees. The rule to follow when running the program is that side A is opposite angle A.

The program contains four defined functions described below, and the maths involved is 'O' level standard - SIN and COSINE rules.

Program Commentary

20	Traps data not forming triangle.
30-60	Arccos, arcsin, degrees to radians and radians to degrees functions.
310-810	Calculates the remaining data after inputting known data.
830-850	These are necessary because the functions return values of angles between +/- 90 degrees, not 0-180 degrees.
860	Tests to see if angles are ok within limit.
870-990	Prints out all data on triangle.
1000-1050	Error trap for inconsistent data.

```
10 REM Trigonometry
20 ON ERROR GOTO 1000
30 DEFFNC(X)=ATN((SQR(1-X*X))/X)
40 DEFFNS(X)=ATN(X/(SQR(1-X*X)))
50 DEFFNR(X)=(X*ATN(1)/45)
60 DEFFND(X)=(X*45/ATN(1))
70 GOSUB 130
80 ON CH GOSUB 310,390,460,560,670,790
90 IF E=1 THEN GOTO 120
100 GOSUB 820
110 GOTO 70
120 END
130 SCREEN0:WIDTH37
140 KEYOFF:COLOR 15,4,7
150 RESTORE 220
160 LOCATE16,0:PRINT"Menu"
170 FOR I=1 TO 6
180 LOCATE4,4+I*2
190 READW$
200 PRINTI;") ";W$
210 NEXT
220 DATA 3 sides,3 angles,2 sides & included angle
230 DATA 2 angles & included side,2 angles & excluded side,End
240 LOCATE 1,18
250 PRINT"Enter number of choice"
260 CH$=INPUT$(1)
270 CH=VAL(CH$)
280 IF CH<1 OR CH>6 THEN GOTO 260
290 PRINTCH;
300 RETURN
310 CLS:LOCATE 1,3
320 PRINT"Input lengths of three sides"
330 INPUT A,B,C
340 AA=FNC((B*B+C*C-A*A)/(2*B*C))
350 BB=FNC((A*A+C*C-B*B)/(2*A*C))
360 CC=FNC((A*A+B*B-C*C)/(2*A*B))
370 AA=FND(AA):BB=FND(BB):CC=FND(CC)
380 RETURN
390 CLS:LOCATE 1,3
400 PRINT"Input angles (degrees)"
410 INPUT AA,BB,CC
420 A=1
430 B=SIN(FNR(BB))/SIN(FNR(AA))
440 C=SIN(FNR(CC))/SIN(FNR(AA))
450 RETURN
460 CLS:LOCATE1,3
470 PRINT"Input 2 sides"
480 INPUT A,B
490 PRINT:PRINT" Input angle"
500 INPUT CC
510 C=SQR(B*B+A*A-2*A*B*COS(FNR(CC)))
520 BB=FNS((B/C)*SIN(FNR(CC)))
530 AA=FNS((A/C)*SIN(FNR(CC)))
540 BB=FND(BB):AA=FND(AA)
```

```
550 RETURN
560 CLS:LOCATE 1,3
570 PRINT"Input 2 angles"
580 INPUT AA,BB
590 PRINT:PRINT" Input side"
600 INPUT C
610 CC=180-AA-BB
620 S=C/SIN(FNR(CC))
630 B=S*SIN(FNR(BB))
640 A=S*SIN(FNR(AA))
650 C=S*SIN(FNR(CC))
660 RETURN
670 CLS:LOCATE 1,3
680 PRINT"Input 2 angles"
690 INPUT AA,BB
700 PRINT:PRINT"Is the side opposite angle a or b?"
710 A$=INPUT$(1)
720 PRINT:PRINT"Input the side"
730 IF A$="a" OR A$="A" THEN INPUT A:GOTO 760
740 IF A$="b" OR A$="B" THEN INPUT B:GOTO 760
750 GOTO 710
760 CC=180-AA-BB
770 IF INSTR("Aa",A$)<>0 THEN S=A/SIN(FNR(AA)) ELSE S=B/SIN(FNR(BB))
780 GOTO 630
790 CLS
800 E=1
810 RETURN
820 CLS
830 IF AA<0 THEN AA=180+AA
840 IF BB<0 THEN BB=180+BB
850 IF CC<0 THEN CC=180+CC
860 IFABS(AA+BB+CC-180)>.5 THEN ERROR
870 LOCATE3,4:PRINT"Side";
880 PRINTTAB(18)"Angle"
890 LOCATE0,6
900 PRINT"A) ";CSNG(A);TAB(18);CSNG(AA)
910 LOCATE 0,9
920 PRINT"B) ";CSNG(B);TAB(18);CSNG(BB)
930 LOCATE 0,12
940 PRINT"c) ";CSNG(C);TAB(18);CSNG(CC)
950 LOCATE 0,15
960 PRINT"Press space bar to continue"
970 Q$=INPUT$(1)
980 IF Q$<>" " THEN GOTO 980
990 RETURN
1000 CLS
1010 LOCATE1,5
1020 PRINT"Data does not form triangle"
1030 PRINT:PRINT:PRINT" Press space bar to return to menu"
1040 Q$=INPUT$(1):IF Q$<>" " THEN GOTO 1040
1050 RUN
```

Pie Chart

Summary

Here we have a routine to produce high resolution pie-charts using numbers entered by the user.

When run the program prints up a set of instructions and will then ask you for the number of quantities you wish to enter to form the pie chart. Type in a number (less than 100 if possible) followed by RETURN. The computer will then attempt to draw the result.

If at any stage the computer tells you that it can't produce an accurate pie chart from the data you have entered try again with either a smaller number of quantities or quantities within a smaller numerical range. When successful a circle should appear divided by lines showing the relative sizes of the quantities you entered.

If the display produced isn't perfectly circular alter the value of AR! in line 110 as follows :

    Reducing AR! - Increases horizontal size of circle.
    Increasing AR! - Reduces horizontal size of circle.

Program Commentary

80-130   Sets data array and variables, prints instructions.

140-150   Define functions to calculate the X & Y coordinates of the points where dividing lines meet circle.

160-190   Get number of entries and input data. N.B. N% isn't range checked so that the maximum amount of data can be entered into the memory space available.

200-330   Main Program Loop :

200-240   Move to centre of circle and add up all entries.

250-280   Set up more variables and draw pie chart circle.

290-330   Loop to draw dividing lines, using functions from lines 140-150.

340-380   Prompt for rerun, wait for space bar to be pressed.

390-450   Error trap routine - inform user of errors.

460-480   Centred text routine (text screen).

490-630   Instruction text.

650-670   Centred hi-res routine (screen 2).

```
10 REM ****************************
20 REM **                        **
30 REM **   Pie Chart Program    **
40 REM **                        **
50 REM ****************************
60 REM
70 REM
80 DIMN!(50)
90 OPEN"GRP:"AS#1
100 KEYOFF:COLOR15,0,0
110 AR!=1.3:REM ADJUST X:Y RATIO TO GIVE CIRCLE - ADJUST VALUE IF NEEDED
120 SCREEN0:GOSUB500
130 ONERRORGOTO400
140 DEFFNX%(A%,R%)=128+R%*SIN(A%/F!)/AR!
150 DEFFNY%(A%,R%)=91+R%*COS(A%/F!)
160 PRINT:PRINT:INPUT"Number of quantities";N%
170 FORA%=1TON%
180 PRINT"Quantity no...";A%;:INPUTN!(A%)
190 NEXT
200 SCREEN2,,0
210 DRAW"BM128,96"
220 T!=0:FORA%=1TON%:T!=T!+N!(A%):NEXT
230 F!=45/ATN(1)
240 S!=0
250 I%=128:E%=186
260 COLOR3,0,0
270 DRAW"BM128,186"
280 CIRCLE(128,91),85,,,,AR!
290 FORA%=1TON%
300 PSET(FNX%(S!,85),FNY%(S!,85))
310 DRAW"M128,96"
320 O!=S!:S!=S!+N!(A%)/T!*360
330 NEXT
340 BEEP
350 P$="Press SPACE BAR to re-run":Y%=183:GOSUB660
360 FORA%=0TO30:I$=INKEY$:NEXT
370 I$=INKEY$:IFI$<>" "THEN370
380 GOTO100
390 REM ** ERROR TRAP **
400 SCREEN0
410 P$="The data you have entered will not":YP%=4:GOSUB470
420 P$="Produce an accurate pie chart.":YP%=6:GOSUB470
430 P$="Please press SPACE and try again":YP%=8:GOSUB470
440 I$=INKEY$:IFI$<>" "THEN440
450 RUN
460 REM ** CENTRED TEXT ROUTINE **
470 LOCATE20-LEN(P$)/2,YP%:PRINTP$;
480 RETURN
490 REM ** INSTRUCTIONS **
500 P$="This program enables you to type in":YP%=0:GOSUB470
510 P$="A set of numbers for the computer to":YP%=1:GOSUB470
520 P$="Turn into a coloured pie-chart.":YP%=2:GOSUB470
530 P$="You will first be asked for the number":YP%=4:GOSUB470
540 P$="Of entries you wish to make":YP%=5:GOSUB470
```

```
550 P$="(Range 1 - 50)":YP%=6:GOSUB470
560 P$="Then you will be asked to type in each":YP%=7:GOSUB470
570 P$="One of your entries.":YP%=8:GOSUB470
580 P$="The computer will then try to produce":YP%=9:GOSUB470
590 P$="A suitable pie chart.But if this is":YP%=10:GOSUB470
600 P$="Not possible you will be asked to":YP%=11:GOSUB470
610 P$="Retry.":YP%=12:GOSUB470
620 P$="When the pie-chart is finished press":YP%=13:GOSUB470
630 P$="SPACE to restart":YP%=14:GOSUB470
640 RETURN
650 REM ** CENTRED TEXT ROUTINE **
660 PRESET(128-LEN(P$)*4,Y%):PRINT#1,P$
670 RETURN
```
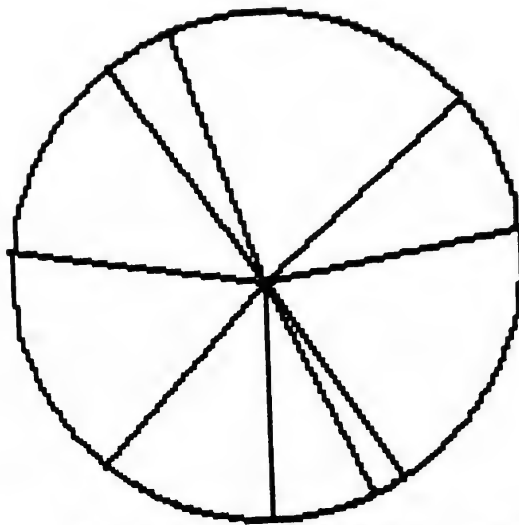

Press SPACE BAR to re-run

Prime Numbers

Summary

This program performs two functions, using the same subroutine for both. It will either test a number for primeness, or list all the primes between two numbers. The primeness testing routine could be speeded up by using integer variables, but this places an upper limit of 32767 on any numbers used.

Primeness is tested simply by counting integers from one to the square root of the number tested. If the number divides exactly by the loop variable then it is not prime and the test ends, returning a value of one in the flag, F.

Program Commentary

20-90   Main control loop - choose option, return to menu.

100-150   Controls test of one number, input by user.

160-180   Tests number N for primeness, sets F=1 if not prime.

190-260   Loops through numbers between limits, printing those that are prime.

```
10 REM Primes
20 SCREEN 0:WIDTH 37:KEYOFF:COLOR 15,4,7
30 LOCATE0,5:PRINT"A) Test for primeness"
40 PRINT:PRINT"B) List Primes"
50 LOCATE 0,10:PRINT"Enter letter of choice"
60 CH$=INPUT$(1):IF INSTR("AaBb",CH$)=0 THEN GOTO 60
70 IF CH$="A" OR CH$="a" THEN GOSUB 100 ELSE GOSUB 190
80 LOCATE 11,0:PRINT"Press space bar"
90 Q$=INKEY$:IF Q$=" "THEN GOTO 20 ELSE GOTO 90
100 CLS:LOCATE0,5
110 F=0:INPUT"Number to test";N
120 IF N<2 THEN GOTO 110
130 GOSUB 160
140 PRINT:IF F=1 THEN PRINT"Not ";
150 PRINT"Prime":RETURN
160 FOR I=2 TO SQR(N)
170 IF N/I=INT(N/I) THEN F=1:I=SQR(N)+1
180 NEXT:RETURN
190 CLS:LOCATE 0,5:INPUT"Enter lower limit";L
200 LOCATE 0,7:INPUT"Enter upper limit";H
210 IF H<L THEN GOTO 190
220 PRINT
230 FOR N=L TO H:F=0
240 GOSUB 160
250 IF F=0 THEN PRINTN;
260 NEXT:RETURN
```

Alphasort

Summary

This program uses a fairly simple sort routine that, while being slower and less efficient than the Shell-Metzner sort in this book, is fairly easy to understand and should give you some idea of the way in which a simple sort routine works.

The routine the program uses is fairly simple and proceeds as follows :

- LAST = 1
    START
        - Search list up to entry number LAST for name nearest end of alphabet.
        - Swap this name with name number LAST.
        (This name is now in the right place)
        - LAST = LAST+1 (Exclude this name from further sorting).
        - LAST equals number of names entered ?
        If not then goto START.
    - Print out list of names last to first to give sorted list.
- Repeat or end as required.

The program allows up to 100 names to be entered (!!), each up to 20 characters long. If either of these conditions is not met or the computer runs out of memory the program will tell you, clear entries made so far and ask you for the number of names again.

Program Commentary

70-80   Set up screen, error trapping and print instructions.

90-100   Enter number of names and check for range.

110-160   Enter the list of names.

170-290   Main Sort Loop :

170-230   Corresponds to algorithm shown above.

240-280   Print out sorted list and inform user/ask for repeat.

320-510   Print out centred instructions.

520-530   Centred text routine.

```
10  REM ****************************
20  REM **                        **
30  REM ** Alphabetic Sort Program **
40  REM **                        **
50  REM ****************************
60  REM
70  ONERRORGOTO300
80  SCREEN0:KEYOFF:GOSUB320
90  PRINT:PRINT:INPUT"How many names ";N%
100 IFN%<1THENPRINT"Not acceptable":BEEP:GOTO90
110 DIMN$(N%)
120 FORA%=1TON%
130 PRINT"Name no....";A%;:LINEINPUTN$(A%)
140 IFLEN(N$(A%))<1THENPRINT"String too short - try again":BEEP:GOTO130
150 IFLEN(N$(A%))>20THENPRINT"String too long - try again":BEEP:GOTO130
160 NEXT
170 FORA%=1TON%
180 B$=CHR$(0):H%=A%
190 FORB%=A%TON%
200 IFN$(B%)>B$THENH%=B%:B$=N$(B%)
210 NEXT
220 SWAPN$(H%),N$(A%)
230 NEXT
240 PRINT:PRINT:PRINT"Names Sorted....":PRINT:PRINT
250 FORA%=N%TO1STEP-1:PRINTN$(A%):NEXT
260 PRINT:P$="Press SPACE to repeat,E to end":P%=24:GOSUB520
270 I$=INKEY$:IFI$<>" "ANDI$<>"E"ANDI$<>"e"THEN270
280 IFI$=" "THENRUN
290 END
300 BEEP:PRINT"Too many entries - Try Again":CLEAR:GOTO90
310 END
320 P$="This program allows the user to enter":P%=1:GOSUB520
330 P$="Up to 100 names of up to 20":P%=2:GOSUB520
340 P$="Characters long for sorting into":P%=3:GOSUB520
350 P$="Alphabetical order.":P%=4:GOSUB520
360 P$="First,enter the number of names you":P%=7:GOSUB520
370 P$="Wish to sort (followed by RETURN)":P%=8:GOSUB520
380 P$="And then type in each name,once again":P%=9:GOSUB520
390 P$="Followed by RETURN":P%=10:GOSUB520
400 P$="When all the names have been entered":P%=11:GOSUB520
410 P$="(Max number of names is 100)":P%=12:GOSUB520
420 P$="The computer will try to sort them into":P%=13:GOSUB520
430 P$="Alphabetic order.If it runs out of":P%=14:GOSUB520
440 P$="Memory during name entry or sorting":P%=15:GOSUB520
450 P$="The computer will stop and tell you":P%=16:GOSUB520
460 P$="This has happened and will return to":P%=17:GOSUB520
470 P$="The start of the program and again ask":P%=18:GOSUB520
480 P$="For the number of names to be sorted":P%=19:GOSUB520
490 P$="Note that a name must be between 1":P%=21:GOSUB520
500 P$="And 20 letters long":P%=22:GOSUB520
510 RETURN
520 LOCATE20-LEN(P$)/2,P%:PRINTP$;
530 RETURN
```

Appendix

Machine Code On The MSX

This appendix is intended as a small extension to the Basic programs. It includes a machine code monitor, which is a utility to facilitate the entry of machine code programs. This is followed by several short machine code routines which can be entered through the monitor, or POKEd directly from a Basic program. The MSX has a number of built-in ROM routines to perform various tasks. As these often involve setting registers and memory locations, they are most easily called from machine code routines, which is how a number of these programs work.

Many of the things shown here could be done in Basic, but would be either slower or less efficient. This appendix does not purport to teach anything about the machine - several books could be written on that alone - but show how easy it can be to perform apparently miraculous tasks such as filling the screen in the blink of an eye.

The routines may be entered as they stand, used in Basic programs or, if you are feeling adventurous, adapted to your own devices. In any case, good luck, and welcome to the world of machine code programming.

Program 1

Get Key Routine

Enter code at : &HC000

Code (in dec) : 205,156,0,194,0,192,205,159,0,33,248,247,119,201

Enter the code from the monitor. Then use DEFUSR=&HC000 to set up the routine. At any time, A=USR(0) will wait for a key to be pressed and return its ASCII code in A.

Program 2

Store Screen 0 Picture

Enter code at : &HD000

Code (in dec) : 17,0,224,33,0,0,1,192,3,205,89,0,201

Program 3

Recall Stored Screen 0 Picture

Enter code at : &HD00D

Code (in dec) : 17,0,0,33,0,224,1,192,3,205,92,0,201

These two programs act together to store the contents of the screen 0 VRAM at memory location &HE00, taking 960 bytes. To store the screen contents, use a USR routine at &HD000. If you

Program 4

Fill Screen

Enter Code at : &HD000

Code (in dec) : 205,108,0,62,65,50,48,208,33,0,0,1,192,3,205,86,
0,205,156,0,32,251,205,159,0,254,81,40,7,33,48,
208,126,60,24,225,201

This really shows off the speed of machine code. Enter the code, and the screen is filled with the letter ´A´. Press a key, and this becomes ´B´, then ´C´ etc. through the character set. Press ´Q´ to end the program.

Machine Code Monitor

Summary

This program is designed to make the entry of machine code a slightly less painful operation than it normally is. While it won't actually solve bugs, it will make searching for them a lot easier. Its facilities include:

1) Type in code : This allows you to enter machine code directly as decimal, octal, binary or hexadecimal.

2) Execute code : This sets up USR1 and calls it, which effectively runs the program.

3) Edit code : This allows any part of memory to be examined and, providing it is in RAM, altered. This should be used for minor alterations and examining code, as major changes are best done with (1) above.

4) Copy bytes : This moves a section of code up or down memory. This is useful when editting as it allows code to be moved up to make space for additional instructions or data. Be careful not to copy programs so that they overwrite themselves, i.e. the destination address should be higher than the end address of the code being copied.

5) Fill with byte : This fills an area of RAM with one number. It can be used to set up data areas, to erase unwanted programs, or to clear areas for programming.

6) Locate a byte : This will give the address of every location within a set limit which contains a specified number. This can be used to locate 'lost' programs or find areas of interest within RAM.

7) Output to port : This simply performs an OUT (N),X .

8) Input from port : Surprise, surprise - performs an INP(X)

9) Save code : This dumps a section of memory to tape.

10) Load code : This loads a BSAVEd section of code from tape into memory.

11) Exit : Allows you to leave the program without performing a CTRL-STOP.

The function keys have also been set up to allow hex, octal, and binary input by providing the necessary prefix with one key push.

Program Commentary

10-30   Clear all variables and screen, set up function keys.

40-200   Display the menu.

210-260   Decides which routine the user requires, and redirects
           control accordingly.

270-380   Option (1).

390-470   Option (2).

480-720   Option (3).

730-800   Option (4).

810-870   Option (5).

880-1010   Option (6).

1020-1050 Option (7).

1060-1100 Option (8).

1110-1180 Option (9). Uses BSAVE to store code.

1190-1250 Option (10).

1260-1290 Exit routine.

```
10 CLEAR200,&H9FFF
20 KEY1,"&h":KEY2,"&b"
30 KEY3,"&o":KEY4,"END"+CHR$(13)
40 SCREEN1,0,0
50 PRINT"M/C MONITER"
60 PRINT:PRINT
70 PRINT" (1)  TYPE IN CODE"
80 PRINT" (2)  EXECUTE CODE"
90 PRINT" (3)  EDIT CODE"
100 PRINT
110 PRINT" (4)  COPY BYTES"
120 PRINT" (5)  FILL WITH BYTE"
130 PRINT" (6)  LOCATE A BYTE"
140 PRINT
150 PRINT" (7)  OUTPUT TO PORT"
160 PRINT" (8)  INPUT FROM PORT"
170 PRINT
180 PRINT" (9)  SAVE CODE"
190 PRINT" (10) LOAD CODE"
200 PRINT
210 PRINT" (11) EXIT"
220 PRINT:PRINT
230 INPUT"WHICH ONE";A$
240 A%=VAL(A$)
250 IFA%>11THENGOTO40
260 ONA%GOSUB280,400,500,750,830,900,1040,1080,1130,1210,1280
270 GOTO40
280 INPUT"START ADDRESS ";ADDRS%
290 PRINT
300 PRINT"   TYPE ´END´ TO FND"
310 PRINT:PRINT
320 LOCATE0,22
330 PRINTADDRS%;"/";HEX$(ADDRS%);" = ";
340 INPUT"CODE >";CODE$
350 IFCODE$="END" THEN RETURN
360 CODE%=VAL(CODE$)
370 POKEADDRA%,CODE%
380 ADDRS%=ADDRS%+1
390 GOTO 320
400 INPUT"START ADDRESS ";ADDRS%
410 PRINT
420 DEF USR1=ADDRS%
430 A%=USR1(0)
440 PRINTA%
450 PRINT
460 PRINT" PRESS ANY KEY TO CONTINUE"
470 K$=INKEY$
480 IFK$=""THENGOTO470
490 RETURN
500 INPUT"START ADDRESS ";ADDRS%
510 PRINT
520 PRINT"USE CURSOR KEYS TO FIND THE"
530 PRINT
540 PRINT"START BYTE THEN HIT SPACE."
```

```
550 PRINT
560 PRINT"    PRESS 'Q' TO QUIT."
570 PRINT
580 B%=PEEK(ADDRS%)
590 PRINTADDRS%;"/";HEX$(ADDRS%);
600 PRINT" > ";
610 PRINTB%;
620 LOCATE25,22
630 IFB%>31THENPRINTCHR$(B%)ELSEPRINT
640 K$=INKEY$
650 IFK$=""THENGOTO 640
660 K%=ASC(K$)
670 IFK%=31THENADDRS%=ADDRS%-1
680 IFK%=30THENADDRS%=ADDRS%+1
690 IFK%=32THENGOTO720
700 IFK%=81THENRETURN
710 GOTO580
720 INPUT"NEW CODE ";CODE%
730 POKE ADDRS%,CODE%
740 GOTO 580
750 INPUT"DESTINATION ADDRESS ";DEST%
760 INPUT"ADDRESS TO COPY FROM";ADDRS%
770 INPUT"NUMBER OF BYTES ";NO%
780 FORA%=ADDRS%TOADDRS%+NO%
790 B%=PEEK(A%)
800 POKEDEST%+A%-ADDRS%,B%
810 NEXTA%
820 RETURN
830 INPUT"START ADDRESS ";ADDRS%
840 INPUT"NUMBER OF BYTES ";NO%
850 INPUT"NEW BYTE ";NB%
860 FORX%=ADDRS%TOADDRS%+NO%
870 POKEX%,NB%
880 NEXTX%
890 RETURN
900 INPUT"START ADDRESS ";START%
910 INPUT"END ADDRESS";E%
920 INPUT"BYTE NUMBER";BYTE%
930 PRINT
940 PRINT"DECIMAL        HEX"
950 PRINT
960 FORA%=START%TOE%
970 IF PEEK(A%)=BYTE%THENPRINTA%,HEX$(A%)
980 NEXTA%
990 PRINT:PRINT
1000 PRINT" PRESS ANY KEY TO CONTINUE"
1010 K$=INKEY$
1020 IFK$=""THENGOTO1010
1030 RETURN
1040 INPUT"PORT NUMBER ";PRT%
1050 INPUT"BYTE ";BYTE%
1060 OUT(PRT%),BYTE%
1070 RETURN
1080 INPUT"PORT NUMBER ";PRT%
```

```
1090 PRINTINP(PRT%)
1100 K$=INKEY$
1110 IFK$=""THENGOTO1100
1120 RETURN
1130 INPUT"START ADDRESS ";START%
1140 INPUT"END ADDRESS ";E%
1150 INPUT"EXECUTION ADDRESS";EXE%
1160 PRINT
1170 INPUT"PRESS RECORD THEN RETURN";R$
1180 PRINT
1190 BSAVE"CAS:CODE",START%,E%,EXE%
1200 RETURN
1210 INPUT"ARE YOU SURE (Y/N) ";YN$
1220 IFYN$="N"THENRETURN
1230 IFYN$="Y"THENGOTO 1250
1240 GOTO1210
1250 PRINT
1260 BLOAD"CAS:CODE"
1270 RETURN
1280 INPUT"ARE YOU SURE (Y/N)";YN$
1290 IFYN$="Y"THEN END
1300 IFYN$="N"THENGOTO10
1310 GOTO1280
```

This ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮ of
fun ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮ d
uti▮▮▮▮▮▮▮▮▮▮▮▮
presented as dire▮▮▮▮ngs from the MSX
and along with each program is both a
summary and a commentary.

The summary describes the program –
what it tries to achieve, the aim of the
game, points arising etc. In the
commentary the program is broken
down into sections, with a full description
of what each part does.

A book packed full of ideas, methods
and concepts for all levels of
programming ability.

6.95net

Published by

Kuma
MSX